SimplIQ
Command Reference Manual

September 2004

Elmo
Motion Control

Important Notice

This guide is delivered subject to the following conditions and restrictions:

- This guide contains proprietary information belonging to Elmo Motion Control Ltd. Such information is supplied solely for the purpose of assisting users of *SimplIQ* servo drives.
- The text and graphics included in this manual are for the purpose of illustration and reference only. The specifications on which they are based are subject to change without notice.
- Information in this document is subject to change without notice.

Doc. No. MAN-SIMCR
Copyright © 2003, 2004
Elmo Motion Control
All rights reserved

Revision History

Ver. 3.0	Sept. 2004	document name changed from Harmonica C.R. to *SimplIQ* C.R.	
		changed commands:	AN, AS, CA, EC, EM, GS, HM, HY, IB, IF, IL, IP, OB, OL, OP, PS, RM, SN, UM, VH, WS, XP
		added commands	OC
		Functional Listing chapter removed	
Ver. 2.1	Sept. 2003	changed commands:	AC, DC, LL, MF, PP, QP, RM, SD, UM, VH, XP
		added command:	SN
		other changes:	limit ranges
Ver. 2.0	May 2003	changed commands:	AC, AG, BP, BT, DC, DV, EC, EM, EO, ET, FR, GS, HX, IF, HV, IL, KV, LL, LP, OL, PK, PP, PR, QP, RC, RG, RL, RM, RV, SP, SR, SV, TC, TM, TS, TW, VH, VR, WI, WS, XP, YA, ZP, ZX
Ver. 1.0	Nov. 2002	Initial Release	***(HARCREN1102)***

Elmo Motion Control Inc.
1 Park Drive, Suite 12
Westford, MA 01886
USA
Tel: +1 (978) 399-0034
Fax: +1 (978) 399-0035

Elmo Motion Control GmbH
Steinbeisstrasse 41
D-78056, Villingen-Schwenningen
Germany
Tel: +49 (07720) 8577-60
Fax: +49 (07720) 8577-70

www.elmomc.com

Contents

Chapter 1: Introduction 1-1
1.1 Command Specification 1-1
1.2 Scope 1-2

Chapter 2: Alphabetical Listing 2-1
Limit Ranges 2-3
AC - Acceleration 2-4
AG[N] - Analog Gains Array 2-5
AN[N] - Analog Inputs Array 2-6
AS[N] - Analog Input Offsets Array 2-7
BG - Begin Motion 2-8
BH - Get a Single Recorded Signal as Hexadecimal 2-9
BP[N] - Brake Parameter 2-10
BT - Begin Motion at Defined Time 2-11
BV - Maximum Motor DC Voltage 2-12
CA[N] - Commutation Array 2-13
CC - Compiled Program Ready 2-18
CD - CPU Dump 2-19
CL[N] - Current Continuous Limitations and Motor Stuck Protection Parameters 2-21
CP - Clear Program 2-23
DD - CAN Controller Status 2-25
DF - Download Firmware 2-26
DL - Download Program 2-27
DV[N] - Reference Desired Value 2-28
EC - Error Code 2-29
EF[N] - Encoder Filter Frequency 2-42
EM[N] - ECAM Parameters 2-44
EO - Echo Mode 2-46
ER[N] - Maximum Tracking Error 2-47
ET[N] - Entries for ECAM Table 2-48
FF[N] - Feed Forward 2-49
FR[N] - Follower Ratio 2-50
GS[N] - Gain Scheduling 2-51
HL[N] - Over-speed Limit and Position Range Limit 2-53
HM[N] - Homing, Capture and Flag 2-54
HP - Halt Program Execution 2-57
HV[N] - Commutation Tables 2-58
HX - Hexadecimal Mode 2-59
HY[N] - Auxiliary Homing, Capture and Flag 2-60
IB[N] - Input Bits Array 2-63
ID, IQ - Read Active Current and Reactive Current 2-64
IF[N] - Digital Input Filter 2-65
IL[N] - Input Logic 2-66
IP - Input Port 2-73
JV- Jogging Velocity 2-75
KG[N] - Gain Scheduled Controller Parameters 2-76
KI[N], KP[N] - PI Parameters 2-77
KL - Kill Motion and Program 2-78

KV[N] - High-order Controller Filter Parameters 2-79
LC - Current Limit Flag 2-80
LD - Load Parameters from Flash 2-81
LL[N] - Low Feedback Limit 2-82
LP[N] - List Properties 2-83
LS - List User Program 2-84
MC - Maximum Peak Driver Current 2-85
MF - Motor Failure 2-86
MI - Mask Interrupt 2-89
MO - Motor Enable/Disable 2-91
MP[N] - Motion (PT/PVT) Parameters 2-93
MS - Motion Status 2-95
OB[N] - Output Bits Array 2-96
OC[N] – Output Compare 2-98
OL[N] - Output Logic 2-101
OP - Output Port 2-103
PA - Absolute Position 2-104
PE - Position Error 2-105
PK - Peek Memory 2-106
PL[N] - Peak Duration and Limit 2-107
PM - Profiler Mode 2-109
PP[N] - Protocol Parameters 2-110
PR - Relative Position 2-112
PS - Program Status 2-113
PT - Position Time Command 2-114
PV - Position Velocity Time Command 2-115
PW[N] - PWM signal parameters 2-116
PX - Main Position 2-117
PY - Auxiliary Position 2-118
QP[N], QT[N], QV[N] - Position, Time, Velocity 2-119
RC - Define Recorded Variables 2-120
RG - Recorder Gap 2-121
RL - Record Length 2-122
RM - Reference Mode 2-123
RP[N] - Recorder Parameters 2-124
RR - Activate Recorder / Get Recorder Status 2-126
RS - Soft Reset 2-127
RV[N] - Recorded Variables 2-128
SD - Stop Deceleration 2-129
SF - Smooth Factor 2-130
SN - Serial Number 2-131
SP - Speed for PTP Mode 2-132
SR - Status Register 2-133
ST - Stop Motion 2-135
SV - Save Parameters to Flash 2-136
SY - Say Something 2-137
TC - Torque Command 2-138
TM - System Time 2-139
TP[N] - Floating Wizard Parameters 2-140
TR - Target Radius 2-141
TS - Sampling Time 2-142
TW[N] - Wizard Command 2-143

UF[N] – User Float array 2-144
UI[N] – User Integer 2-145
UM - Unit Mode 2-146
VE - Velocity Error 2-148
VH[N], VL[N] - High and Low Reference Limit 2-149
VR - Firmware Version 2-150
VX, VY - Velocity of Main and Auxiliary Feedback 2-151
WI[N] - Miscellaneous Reports, Integer 2-152
WS[N] - Miscellaneous Reports 2-153
XA[N] - Extra Parameters (more) 2-156
XC, XQ - Execute or Continue Program 2-157
XM[N] - X Modulo 2-158
XP[N] - Extra Parameters 2-159
YA[N] - Auxiliary Position Sensor Parameters 2-161
YM[N] - Y Modulo 2-163
ZP[N] - Integer Wizard Parameters 2-164
ZX[N] - User Program and Auto-tuning Temporary Storage 2-165

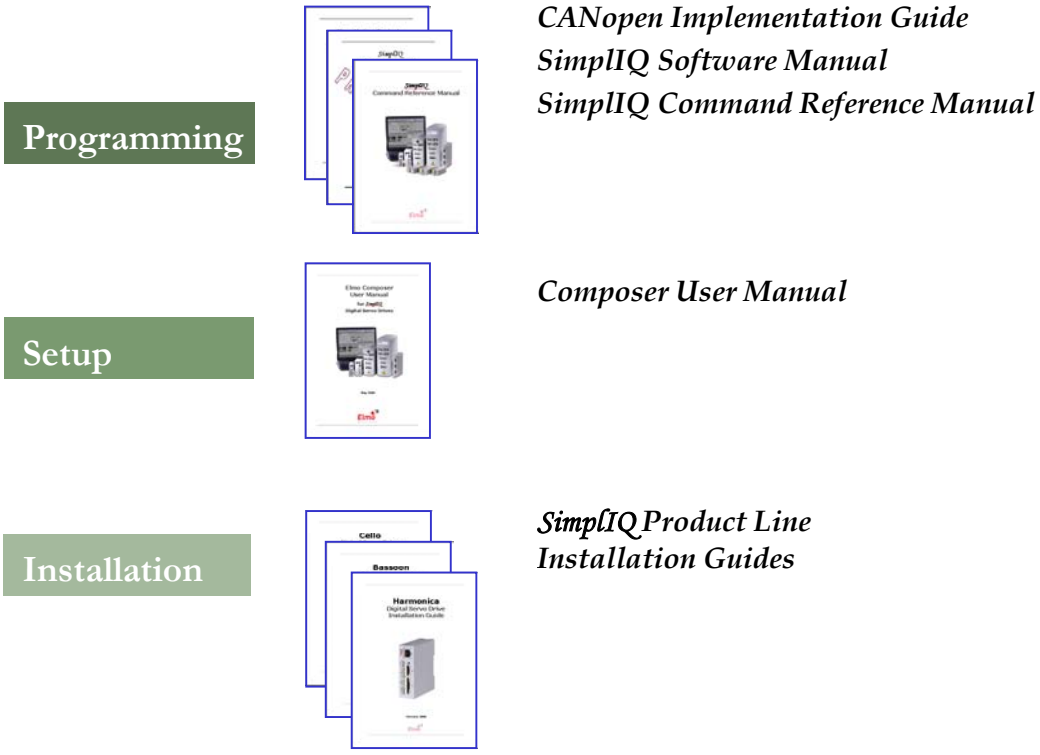
Index I-1

Chapter 1: Introduction

This manual describes, in detail, each software command used to manipulate the *SimplIQ* line of digital servo drives. It is an integral part of the *SimplIQ* documentation set, which includes:

- The Harmonica, Cello and Bassoon *Installation Guides,* which provides full instructions for installing a drive
- The Composer *User Manual,* which includes explanations of all the software tools that are a part of Elmo's Composer software environment
- The *SimplIQ Software Manual,* which describes the comprehensive software used with the *SimplIQ* line of digital servo drives

The following diagram illustrates the hierarchy of *SimplIQ* documentation.



1.1 Command Specification

Commands for *SimplIQ* drives may be specified from the following sources:

- User program — A program loaded to the servo drive via one of the communication options. After program execution begins, the program is managed by the drive.
- RS-232 — Serial, point-to-point, short-range communication. Although this method is rather slow, RS-232 is very easy to use and requirements are minimal: a standard PC with serial port and ASCII terminal software.

- CANopen — Serial, multi-drop, medium speed and medium-range communication. This type of communication requires special-purpose host hardware and software.

This manual describes the *SimplIQ* commands that can be specified from each of these sources. Most of the commands are equally available for all three sources. Certain commands, however, are limited in scope according to type of program or communication.

All the commands are available to CAN communication in text form through the OS service, objects 0x1023 and 0x1024. In addition, the numerical set/get commands are available to CAN users in short PDO form, called the "binary interpreter." The binary and the OS SCAN interpreters are described fully in the CAN manual.

CANopen may also be used to manipulate the drive using the object dictionary (OD) method, which is the native CAN method. This manual does not cover OD manipulations with CANopen; refer to the "Object Dictionary" section of the CANopen manual for full explanations.

The *SimplIQ* drive responds to many privileged commands — such as those used by the Composer setup wizard — that are not documented in this manual.

1.2 Scope

This manual includes the complete list of commands used by *SimplIQ* servo drives. It specifies how to use each command, along with added remarks and examples.

The commands are presented in two ways:

- A task-related listing
- Alphabetically

In the task-related reference, the commands are sorted into groups of related commands. Each group is presented in a table listing the commands with basic descriptions. The alphabetical command listing provides a detailed explanation of each command, with examples and references to the *SimplIQ Software Manual* when necessary.

This *Command Reference Manual* does not cover the following topics:

- User program keywords, used for writing user programs. These, as well as other issues of developing, running and debugging user programs, are covered in the *SimplIQ Software Manual.*
- Interpreter functions and operators. The *SimplIQ* interpreter allows complex arithmetic expressions and supports many arithmetic, trigonometric and logical operators. The syntax for interpreter commands is explained in the "Interpreter Language" chapter of the *SimplIQ Software Manual.*

Chapter 2: Alphabetical Listing

This chapter lists all the commands in alphabetical order, along with detailed definitions and examples of each command.

The description of each command includes the following items:

Purpose: The operation or task of the command

Attributes: The characteristics of the command

Type: One of the following:

- A **command**: An instruction to do something. For example, the BG (Begin Motion) command starts a new motion profile.
- A **parameter**: A data item that may be used later. For example, the AC (Acceleration) parameter is required for calculating subsequent motions.
- A **status report**: Get a value, such as the motor speed, a digital input or the reason for the last motor failure.

The parameters and certain commands have numerical values, as follows:

- **Integer**: A 32-bit long integer
- **Real**: A 32-bit floating point number (IEEE style)
- **String**: A set of printable ASCII characters

Integer variables may have the following attributes:

- **Bit field**: The integer should be understood not as a number but rather as a combination of binary fields. For example, the IP (Digital Input) command reads many On/Off switches to the same integer, allocating one bit for each.
- **Option**: A selector that may accept one of several options. For example, the motor direction may be set to forward or reverse, symbolized by the numbers 0 and 1 respectively.

Source: Defines the "agents" that may use the command, as follows:

- RS-232 communication
- CANopen communication
- User program

The command access rights are not equal for all sources. For example, CANopen binary interpreter cannot use the string commands listed in this manual. Another example is the LS (List Program) command that, of course, cannot be performed from within a program.

Restrictions: The use of certain commands is illegal in certain contexts. The reasons for this may be:

- *Safety:* For example, it is not safe to change the direction of the feedback while the motor is running.
- *Relevance:* For example, a torque command cannot be specified in speed control mode (UM=2); in speed mode, the drive automatically sets the torque.
- *Consistency:* A parameter may be inconsistent with the specification of other parameters. For example, in point-to-point mode, the position absolute value (PA) may be no higher than the maximum allowed position reference (VH[3]).
- *Product grade:* Elmo drives come in Standard and Advanced grade (model). When no product grade restriction is cited, the command is relevant for both grades.

Default values: Default value and storage type.
Volatile variables are reset to their defaults with each power on. Non-volatile variables can be stored using the SV command. Stored non-volatile values are read from storage upon power on and can be reset to their defaults using the RS command.

Range: Range definition: For example, the position command may be specified in the range [-1,073,741,824…-1,073,741,823]

Unit mode (UM): Defines the function of the drive. The unit modes are:

- UM=1 Torque control
- UM=2 Speed control
- UM=3 Micro-stepping
- UM=4 Dual-feedback position control
- UM=5 Single-feedback position control

Activation: Specifies when the entered parameter value should be used.
Activation may be:

- Immediate As soon as the command is processed
- Triggered By another command

For example, the AC (acceleration) parameter should only affect the next motion, triggered by the BG command.

Examples: Simple examples of the command usage. All examples are given in RS-232 syntax.

See also: Related commands

Reference chapter in the *SimplIQ Software Manual:* Chapter or section that contains relevant details pertinent to the command.

Limit Ranges

The following table lists the value ranges for defining the limits of the system.

Subject	Values
Position counter ranges	Main position counter is subjected to a modulo counting with the following ranges: XM[1]: Lowest value XM[2]: Highest value Range: [-10^9…10^9] counts Auxiliary position counter is limited to: YM[1]: Lowest value YM[2]: Highest value Range: [-10^9…10^9] counts
Velocity range	Range for **Quadrature Encoder**: [-20,000,000…20,000,000] counts/sec Range for **other feedbacks**: [-80,000,000…80,000,000] counts/sec EF[1]: Filter for main velocity sensor EF[2]: Filter for auxiliary velocity sensor
Acceleration/Deceleration ranges	Range: [100…1,000,000,000]
Stop deceleration range	Range: [400…1,000,000,000]
Torque limits	Range of torque command is subjected to the following limits: CL[1], PL[1] Range: [-MC…MC] Note: ▪ The torque in RM=1 is taken as a summary of the external and software reference. ▪ The multiplication of the PWM frequency reduces the torque limit.

AC - Acceleration

Purpose:
Defines the maximum acceleration in counts/second^2. This parameter is used in speed mode (UM=2) and position control modes (UM=3, 4, 5) in PTP (PA, PR) and jogging (JV) reference modes.

The AC parameter does not affect the present motion. It is used for planning the next motion, which is initiated by a BG command.

If AC is smaller than SD, the maximum possible acceleration will be limited to SD and the value of AC will be ineffective.

Attributes:	**Type:**	Parameters, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	20,000,000 (RS), Non-volatile
	Range:	Acceleration range
	Unit modes:	UM=2, 3, 4, 5
	Activation:	BG for RM=0, MO=1 for RM=1

Typical applications:
1. Define acceleration limits for the motion (UM=2)
2. Plan a profiled motion (UM=3, 4, 5)

See also:
DC, SP, SV, PA, PR, BG

Reference chapter in the *SimplIQ Software Manual:*
Chapter 12, "The Position Reference Generator"

AG[N] - Analog Gains Array

Purpose:
Sets the gains for preconditioning analog signals, when RM = 1:

- AG[1] sets the gain of analog input #1 when used as a torque command (UM=1, 3).
- AG[2] sets the gain of analog input #1 when used as a speed command (UM=2).

When RM = 0, the AG[N] parameters are ignored.

The meaning of the analog gains depends on the unit mode, as shown in the following table.

Value	Description	Units
UM=1, 3	One volt at the analog reference input command controls the motor phase current of AG[1] amperes.	Ampere/volt
UM=2	One volt at the analog reference input command controls a speed of AG[2] counts/second.	Count/second/volt

Table 2-1: **Analog Gains - Analog Input #1**

Notes:

- AG[1]defines motor phase amperes and not RMS amperes.
- In stepper mode (UM=3), the two external inputs play different roles: The analog input voltage sets the motor power while the follow pulse/direction or quadrature input determines the position.
- The polarity of the analog reference signal may be reversed by setting the sign of the AG[N] parameter accordingly.

Attributes:

Type:	Parameter, Real
Source:	Program, RS-232, CANopen
Restrictions:	None
Default values:	AG[1]=0.1 AG[2]=1 Non-volatile
Range:	[-10,000,000…10,000,000]
Index range:	[1, 2]
Unit modes:	All
Activation:	Immediate

See also:
AN[N], AS[N], UM, RM, VH[N], VL[N]

Reference chapters in the *SimplIQ Software Manual:*
Chapter 11, ”Unit Modes;“ Chapter 10, “The Current Controller”

AN[N] - Analog Inputs Array

Purpose:

- AN[1] reports the analog input #1 value after offset correction (AS[1]), in volts.
- AN[2] reports the analog input #2 value after offset correction (AS[2]) in volts.
- AN[3] reports the measured current in the motor A phase, in amperes.
- AN[4] reports the measured current in the motor B phase, in amperes.
- AN[5] reports the measured current in the motor C phase, in amperes.
- AN[6] reports the line voltage value, in volts.
- AN[7] reports the duty cycle value of the PWM signal after offset correction in fractional units

Attributes:	**Type:**	Status report, Real
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Unit modes:	All

Typical applications:
1. Reading external sensors that provides +/- 10V
2. Reading analog or PWM references for either velocity or current
3. Reading phase currents and line voltage

Note:

- Analog input #1 serves as reference input for analog torque or analog velocity command while in auxiliary reference mode (RM=1).
- AN[7] is available after the first MO=1.
- Each of Elmo's *SimplIQ* drives support a different number of analog inputs. For specific details consult the drive's *Installation Guide*.

See also:
AG[N], AS[N], PW[N]

AS[N] - Analog Input Offsets Array

Purpose:
Compensates for offsets of the analog signals, which may be caused by the limited precision of the SimplIQ electronics.

At times, the signals at the A/D converter may be offset: that is, the A/D reading may be non-zero when a zero reading is desired. This offset may disturb normal operation. An offset reference or feedback signal may cause a motor to "crawl" when a complete stop is desired.

The analog offset subtracts from the analog input as follows:
Corrected signal = A/D reading – Analog offset

- AS[1] - Analog input offset command for analog input #1, in volts
- AS[2] - Analog input offset command for analog input #2, in volts

Attributes:		
	Type:	Parameter, Real
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	0 (RS), Non-volatile
	Range:	[-10.0…10.0] *5 mV resolution*
	Unit modes:	All
	Activation:	Immediate

Note:

- To null the input offsets of the drive, short the analog inputs to ground. Then set AS[1] = AN[1] and AS[2] = AN[2] for analog input #1 and #2 respectively.
- Each of Elmo's *SimplIQ* drives support a different number of analog inputs. For specific details consult the drive's *Installation Guide*.

See also:
AG[N], AN[N]

BG – Begin Motion

Purpose:
Immediately starts the next programmed motion.

- In ***software speed mode*** (UM=2), BG activates the latest JV, and also the new smooth factor (SF), acceleration (AC) and deceleration (DC).
- In ***stepper or position mode*** (UM=3, 4 or 5), BG starts the latest position mode programmed: a point-to-point motion (PA), a jogging motion (JV) or any type of tabulated motion (PVT or PT).

Each motion mode starts with its entire set of parameters. For example, starting a point-to-point motion activates the present of acceleration (AC), deceleration (DC), smooth factor (SF) and speed (SP).

The BG command may be used to *modify* the parameters of the present mode, and not only to program new modes. For example, a BG command in point-to-point mode modifies the active AC parameters (and all other active motion parameters) with its last programmed value.

A "hardware BG" can be accepted via one of the digital inputs (refer to the IL[N] command).

Attributes:	**Type:**	Command, No value
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=1
	Unit modes:	UM=2, 3, 4, 5
	Activation:	Immediate

Notes:

- In position mode (UM=3, 4, 5), BG does nothing if a motion mode (JV, PA, PV, PT) was not previously submitted.
- In "Quick stop" mode (refer to the Elmo *CANopen Implementation Guide*), BG is blocked and returns an error. "Quick stop" mode can be command controlled by a CAN master using the DS402 standard control word (object 0x6040).

See also:
IL[N], BT

BH - Get a Single Recorded Signal as Hexadecimal

Purpose:
Uploads the values recorded by the recorder to a host. The BH command is designed to optimize data transfer from the drive to the host, assuming that the host has the computing power to analyze the drive message.

Attributes:	**Type:**	Command, Integer, Bit-field
	Source:	RS-232
	Restrictions:	RR=0 (valid recorder data is ready), Not while executing a previously-requested BH=n command
	Unit modes:	All
	Activation:	Immediate

Reference chapter in the *SimplIQ Software Manual:*
Chapter 7.4, "The Recorder"

BP[N] - Brake Parameter

Purpose:
Defines the timing of the brake system in the motor when at least one of the digital outputs has been defined by the OL[N] command as a brake. *For safety reasons, a brake-active output releases the brake so that the brake is activated when the drive is not powered on. The brake output is always defined as active low.*

When the brake is released at motor start (MO=1), the drive allows the brake time to disengage before motion begins. During this time, the drive keeps the motor in its starting position. When the motor is turned off (MO=0), the drive first commands the brake to engage. Then, for a time, it keeps the motor in place while the brake actually engages.

- BP[1] - Defines the delay for engaging the brake after the motor is disabled (msec)
- BP[2] - Defines the delay required to disengage the brake after the motor is enabled (msec)

Notes:

- If the motor is disabled by an emergency in real time, the brake is activated at the instant the motor is disabled. The motor freewheels until the brake is fully engaged.
- Response time to interpreter commands (from the user program or communication) is extended during motor disable (MO=0) and enable (MO=1) in BP[1] and BP[2] milliseconds respectively.
- Automatic phasing (commutation search with no digital Hall sensor or other absolute position sensor) is not recommended for a system that requires brake activation.

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=0
	Default values:	BP[1]=0 BP[2]=0 Non-volatile
	Range:	BP[1]: [0…500] BP[2]: [0…500]
	Index range:	[1, 2]
	Unit modes:	All
	Activation:	Immediate

See also:
OL[N], OP

Reference chapter in the *SimplIQ Software Manual:*
Chapter 14, "Limits, Protections, Faults and Diagnosis"

BT - Begin Motion at Defined Time

Purpose:
Starts motion at the defined time. This command is designed to start the simultaneous motion of several axes. It is similar to the BG command with the following difference: BG starts motion immediately whereas BT begins at the defined time.

Syntax:
BT=N
where N is the absolute time in microseconds

Attributes:	**Type:**	Command, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=1
	Unit modes:	UM=2, 3, 4, 5

See also:
BG, TM

BV – Maximum Motor DC Voltage

Purpose:
Reports the scale factor for the drive bus voltage, in volts. This command indicates the type of power amplifier hardware.

Attributes:	**Type:**	Status report, Integer
	Scope:	Program, RS-232, CANopen
	Restrictions:	None
	Unit modes:	All
	Activation:	Immediate

See also:
XP[N]

CA[N] - Commutation Array

Purpose:
Defines motor and commutation parameters. The CA[N] array includes the parameters of the initial motor setup. The CA parameters need to be clearly defined in order to ensure that the motor rotates at all, and so that the feedback direction is correct.

The CA[] array is typically programmed by the Elmo Composer during system configuration (by the wizard). It is not recommended to modify these parameters manually.

Setting of any of the following parameters causes the commutation sequence to be searched again in systems with no digital halls.

The parameters in the following tables define the location and polarity of the Hall sensors and encoder.

Command	Description
CA[1]	Digital Hall sensor A polarity (1 for active high, 0 for active low).
CA[2]	Digital Hall sensor B polarity (1 for active high, 0 for active low).
CA[3]	Digital Hall sensor C polarity (1 for active high, 0 for active low).
CA[4]	Actual Hall sensor connector to Hall A connector pin: 1 for A, 2 for B and 3 for C.
CA[5]	Actual Hall sensor connector to Hall B connector pin: 1 for A, 2 for B and 3 for C.
CA[6]	Actual Hall sensor connector to Hall C connector pin: 1 for A, 2 for B and 3 for C.
CA[7]	Offset of digital Hall sensors. This parameter compensates for Hall sensor location inaccuracies.

Table 2-2: CA Vector - Digital Hall Sensor Parameters

The parameters in the table that follow are required for Analog Encoder and Resolver signal scaling:

Command	Description
CA[9]	Relative phase of the analog encoder sinusoidal signals, in 65,536 units per cycle. In most systems, CA[9] will fall in the range of [-2048…2048].
CA[11]	Offset for the A (sine) channel of the analog encoder (resolver). The offset is given in ADC units in the range of [-4500…4500].
CA[12]	Offset for the B (cosine) channel of the analog encoder (resolver). The offset is given in ADC units in the range of [-4500…4500].
CA[13]	Relative gain of the A (sine) channel of the analog encoder (resolver) with respect to the B (cosine) channel in the range of [20000…40000]

Table 2-3: - Analog Encoder and Resolver Scaling

CA[15] See below

Command	Description
CA[16]	Feedback direction: 0: Use feedback reading as is 1: Invert the direction of the feedback reading Changing CA[16] does not affect the present position count. Direction changes only when counting future feedback pulses.
CA[17]	Commutation sensor: 1 Commutation derives from main feedback; 0 commutation derives from the motor DHall sensors.
CA[18]	Feedback bits ("counts") per revolution, after resolution is multiplied by 4, in the range [6…530,000,000]. - For **Standard Incremental Encoders** with 1000 lines, CA[18] is 4000. If the motor is linear, CA[18] reflects the electrical cycle. For example, if the encoder has 1000 lines/m (4000 counts/m) and the distance between pole sets is 0.1 m, then CA[18] is 400. In this example, CA[18] could be set as any multiple of 400, such as CA[18]=800 or CA[18]=1200. - For systems with **Halls only**, CA[18] is calculated as CA[19] * 6. - For **Analog Encoders**, the resolution is multiplied by $2^{CA[31]}$. For example, if the analog encoder has 2000 lines and CA[31] is 8, then CA[18] will be 512,000. - For **Resolver** feedback CA[18] is calculated as $2^{16 - CA[34]}$ * Resolver pole pair. For example, if the Resolver resolution is 10 bits and has 1pole pair then CA[34]=16-10=6, and CA[18]= 2^{16-6}=1024. - For **Tachometer** feedback the resolution is statically set to 65,536. For **Potentiometer** feedback the resolution is statically set to 28,000.

Table 2-4: CA Vector - Feedback Setup Parameters

The parameters in the tables that follow are required for commutation setup.

Command	Description
CA[19]	Number of motor pole pairs [1…50]. The number of feedback counts per electrical revolution is CA[18]/CA[19]. For good commutation, the number of feedback counts per electrical rotation should be at least 36. Practically, there is no high limit to the number of encoder lines per pole pairs. It is a long integer number that the drive ultimately modulates to 1024.
CA[20]	Digital Hall sensors present: 0: No digital Hall sensors are connected. If the commutation angle is not yet known, then at motor on, a commutation search will be made. No digital Hall input consistency checks will be made. 1: Digital Hall sensors are connected. Upon motor on, commutation will be performed according to the digital Hall sensors. Continuous encoder-based commutation will begin when the first Hall edge is identified. The drive performs commutation checks by continuously comparing the encoder-derived commutation angle with the Hall sensor recorded status.
CA[21]	Position sensor present: 0: Ignore the position sensor inputs. Commutation will be based on the digital Hall sensors only. 1: The position sensor will be used for commutation.
CA[22]	Main feedback type: 0: Reserved. 1: Main feedback entry used as input from a Resolver 2: Main feedback entry used as input for the quadrature incremental encoder signals. 3: Main feedback entry used as input for analog sine\cosine signal. 4: Main feedback entry used as input for Tachometer signal (Maximum Tachometer signal is ±20V) 5: Main feedback entry used as input for Tachometer signal (Maximum Tachometer signal is ±50V) 6: Main feedback entry used as input for digital halls signals 7: Reserved 8: Main feedback entry used as input for Potentiometer feedback
CA[25]	Motor direction: 0: Reverse phase driving so that the motor direction with positive torque command is reversed. 1: Keep the original motor direction, as connected by user.
CA[28]	DC motor: 0: Standard brushless motor. 1: DC motor — do not perform commutation. Current will flow continuously from the A motor connector pin of the servo drive to the B terminal. The C terminal conducts no current.

Table 2-5: CA Vector - Commutation Setup Parameters

Command	Description
CA[15]	Signal frequency for "No Hall" commutation search process. This signal is derived from the sampling time, according to the following: Time = 128 * TS * $2^{CA[15]}$ * 1e-6. The frequency is 1/Time Hz.
CA[24]	The minimum motor movement to perform result analysis, in counts. When this variable is too low, the commutation search process might fail (MF=0x10,000).
CA[26]	Starting torque for motor-on commutation search process in percentages. Starting torque = (CA[26]/100))*CL[1]

Table 2-6: CA Vector - Automatic Commutation Search Parameters, no Hall Sensors

Command	Description
CA[27]	Maximum acceptable number of iterations for auto-phasing process. If the process fails due to overload (motion amplitude is less than CA[24]), the auto-phasing may be repeated CA[27] times, with the current being doubled at every iteration. If the peak current (PL[1]) is reached at any attempt, the auto-phasing process will stop even if CA[27] allows more iterations.

Table 2-7: CA Vector - Auto-phasing Parameters

Command	Description
CA[23]	Counts per meter (any positive integer): 0: Rotary motor 1: Counts per meter in a linear motor This parameter is not used directly by the drive but is rather stored there for the convenience of the host.

Table 2-8: CA Vector - Miscellaneous Parameters

CA[31]	Resolution for one cycle of the analog signal. Analog encoder cycles at one revolution x $2^{CA[31]}$ counts/rev. CA[31] is in the range [2..12]. Changing CA[31] resets the position counter. For linear motors the resolution is per meter

Table 2-9: - Resolution of the Analog Encoder

The following parameters are required for Resolver operation:

CA[32]	Time delay between reference output and the sampling of the signals: Delay = TS/2 - CA[32] / 40 in [us]. This value is used to produce the reference signal to the Resolver.
CA[33]	Resolver filter frequency in [Hz], must correspond to KV[76]-KV[84] setting. Range 300 ... 1300 Hz.
CA[34]	Configure resolver bits in the range 10..16. The resolver readout resolution is $2^{16-CA[34]}$ bits/resolver cycle. For example, if CA[34]=4 the Resolver reads 4096 bits per cycle. If the Resolver has one pole pair, this will also be the bit count per mechanical revolution. If the Resolver has 2 pole pairs, there will be 8092 counts/mechanical revolution and so on.

Table 2-10: - Resolver Parameters

The following parameters are require for Tachometer operation:

CA[35]	The value of the Tachometer offset in Volts.
CA[36]	The value of the Tachometer gain (from the Tachometer's data sheet) in Volts/kRPM.

Table 2-11: - Tachometer Parameters

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=0
	Default values:	CA[1]=CA[2]=CA[3]=1,CA[4]=3, CA[5]=2, CA[6]=1, CA[13]=32768, CA[15]=4, CA[16]=1, CA[17]=1,CA[18]=4096, CA[19]=2, CA[20]=1, CA[21]=1, CA[22]=2, CA[24]=5, CA[25]=1, CA[26]=20, All other CA parameters are 0 (RS), Non-volatile
	Range:	As defined in the previous tables
	Unit modes:	All
	Activation:	Immediate

Notes:

- The CA parameters are usually set automatically by the Composer program. Avoid setting the CA[N] parameters manually unless you are sure of what you are doing.
- Unused indices are reserved for compatibility with other drive models.
- CA[7], CA[35] and CA[36] are *floated*

See also: MO, UM

Reference chapter in the *SimplIQ Software Manual:* Chapter 9, "Commutation"

CC - Compiled Program Ready

Purpose:
Serves as the last stage of the user program downloading process, verifying a downloaded user program and marking it "ready for use."

The CC=N command specifies the program checksum. If this value coincides with the actual program checksum, the "Program ready" internal flag is set on. Otherwise, an error is returned. The CC query returns 0 if no active program is present, and 1 if the "Program ready" internal flag is set on.

Attributes:	**Type:**	Parameter, Integer
	Source:	RS-232, CANopen
	Restrictions:	MO=0, Program not running
	Range:	[0…2^{32}]
	Unit modes:	All
	Activation:	Immediate

The execution of a CC=N command may take a significant amount of time, approximately 1 second.

Reference chapter in the *SimplIQ Software Manual:*
Chapter 6, "Program Development and Execution"

CD – CPU Dump

Purpose:
Returns the status of the CPU and the database. Call CD if:

- The SR report indicates a CPU exception.
- The MF report indicates a CPU exception.
- An attempt to start the motor returns a "Bad database" error code.

The CD report returns a string similar to the following:
Null Address=0
Failure Address=0
Called Handler=none
Database Status:
Database OK

where:

- "Null Address" is the code address at which a CPU exception occurred. A "0" indicates a normal condition.
- "Failure Address" is the code address at which a stack overflow has occurred. A "0" indicates a normal condition.
- "Called Handler" is the type of CPU exception that occurred. A "none" indicates a normal condition.
- "Database Status" indicates if the recent database check at MO=1 — at power up or during a save (SV) — revealed a consistent database. "Database OK" indicates the normal condition.

Attributes:	**Type:**	Status report, String
	Source:	RS-232
	Restrictions:	None
	Unit modes:	All

Notes:

- If an LD command fails, CD reports the reason for the failure by adding the string "Couldn't load from serial flash" followed by the reason for the failure.
- The 3-second time constant is used because almost every motion system applies high torques for short acceleration periods at low speeds.
- The minimum current limit is MC/128. If CL[1] < MC/128, the CL[1] value will be accepted, but the actual current value will be limited to MC/128.

Example:
Null Address=0
Failure Address=0
Called Handler=none
Database Status:
CA[4], error code=37

This CD report indicates that the database is inconsistent because two of the parameters CA[4], CA[5] and CA[6] are equal.

See also:
MF, SR, MO, EC

CL[N] - Current Continuous Limitations and Motor Stuck Protection Parameters

Purpose:
Defines the continuous loading of the drive.

- CL[1] defines the maximum allowed continuous motor phase current, in amperes. This parameter is used to protect the motor from over-current, and the load from excessive torques. The motor current (torque) command is normally limited to its peak limit, as defined by PL[1]. After a short period of torque demands higher than CL[1] (as defined by the PL[2] parameter and equations in the SimplIQ *Software Manual*), the torque command limit is decreased to CL[1]. The torque command remains limited to CL[1] until the average torque demand falls below 90% of CL[1] for a few seconds. CL[1] has no effect if CL[1] > PL[1].
- CL[2] and CL[3] define how the motor stuck protection is handled. A stuck motor is a motor that does not respond to the applied current command, due to failure of the motor, the drive system or the motion sensor.
  CL[2] defines the tested torque level as a percentage of continuous current limit CL[1].
  CL[3] states the absolute threshold main sensor speed under which the motor is considered not moving. If the motor is stuck, motion fault MF=2,097,152 (0x200,000) is set.
  If CL[2] < 2, the motor stuck protection is not applied.
  For other values of CL[2], the motor is disabled and MF is set to 0x200000 if the motor current command level exceeds a selected level for more than 3 seconds, without the result of a significant motor speed, as defined by CL[3].

Attributes:	**Type:**	Parameter, Real
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	CL[1]=0 (RS), Non-volatile, CL[2]=0 (RS), Non-volatile, CL[3]=60 (RS), Non-volatile
	Range:	CL[1]: [0…MC/2] CL[2]: [0…100] CL[3]: [0…16,000]
	Index range:	[1…3]
	Unit modes:	All
	Activation:	Immediate

Example:
If CL[2]=50 and CL[3]=500, the drive will abort (reset to MO=0) with motion fault (MF) 0x200000 if the torque level is kept at least 50% of the continuous current, while the absolute value of the shaft speed does not exceed 500 counts/sec for a continuous 3 seconds.

Notes:

- The motor stuck protection is always applied to the main sensor. In dual loop applications, this protection does not pertain to failures in the auxiliary sensor.
- The time constant of 3 seconds is taken because almost every motion system applies high torques for short acceleration periods while the speed is slow.
- The minimum current limit is MC/128. If CL[1] < MC/128, the CL[1] value will be accepted, but the actual current value will be limited to MC/128.

See also:
LC, MC, PL[N], TC, MF

Reference chapters in The SimplIQ *Software Manual:*
Chapter 10, "The Current Controller;" Chapter 14, "Limits, Protection, Faults and Diagnosis"

CP – Clear Program

Purpose:
Clears the entire user area in the serial flash memory. The CP instruction must be used before any attempt to write a new program to the drive.

Attributes:		
	Type:	Command, No value
	Source:	RS-232, CANopen
	Restrictions:	MO=0, Program isn't running
	Unit modes:	All
	Activation:	Immediate

Notes:

- CP command execution may take a significant amount of time.
- Writing to the same flash location without setting CP will cause a write failure. and the flash contents will become undefined.

DC - Deceleration

Purpose:
Defines the maximum deceleration in counts/seconds^2. This parameter is used in profiled speed control mode (UM=2, PM=1) and in position point-to-point (PA, PR) and jogging (JV) motions (UM=3, UM=4 and UM=5). The DC parameter does not affect the present motion. It is used to plan the next motion, initiated by a BG command.

Attributes:		
	Type:	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	20,000,000 (RS), Non-volatile
	Range:	Deceleration range
	Unit modes:	UM=2, 3, 4, 5
	Activation:	BG for RM=0, MO=1 for RM=1

See also:
AC, SP, PA, PR, BG

Reference chapter the *SimplIQ Software Manual:*
Chapter 12, ”The Position Reference Generator”

DD – CAN Controller Status

Purpose:
Returns the status of the CAN controller as a string in hexadecimal form without a "0x" prefix. DD is valid only for drives that support CAN controllers.

Call DD if you:

- Suspect that the CAN controller is in Bus Off (no communication) mode
- Suspect that there are many error frames on the CAN bus
- Wish to monitor the CAN controller error activities

The DD command reflects object 0x2082 (refer to the Elmo *CANopen Implementation Guide* for more information).

DD value reports:

- CAN receiver flag, indicating the following states:
  - Overrun
  - Bus off
  - Transmitter error
  - Receiver error
  - Transmitter warning
  - Receiver warning
- CAN receive error counter, reflecting the status of the MSCAN receive error counter
- CAN transmit error counter, reflecting the status of the MSCAN receive error counter
- Network status, which may be one of the following values:
  - 1: Disconnected
  - 2: Connected
  - 3: Preparing
  - 4: Stopped
  - 5: Operational
  - 127: Pre-operational

All data is received from the hardware.

Attributes:	**Type:**	Status report, String
	Overloaded:	No
	Source:	RS-232, CANopen
	Restrictions:	None
	Unit modes:	All

DF – Download Firmware

Purpose:
Downloads a new firmware version. This command is designed as part of a sequence that is normally controlled by the Composer program, which reads the firmware update file provided by Elmo, and performs a sequence of actions that includes the DF command.

The DF command is password protected, in order to avoid accidental calls.

Notes:

- After new firmware is downloaded, the drive reboots. All data stored in temporary variables in the RAM is lost.
- Loading new firmware does not normally affect the non-volatile application variables in the data flash memory. Downloading a major software revision may, however, destroy the non-volatile data. The Composer program will warn you of risks of non-volatile data losses.

Attributes:	**Type:**	Command
	Source:	RS-232 only
	Restrictions:	MO=0, Program not running
	Unit modes:	All
	Activation:	Immediate

DL – Download Program

Purpose:
Downloads data to the serial flash memory of the drive. The DL command is used primarily to download compiled user programs to the drive.

The format of a DL command is:
DL##[hex binary data][esc]checksum]

Notes:

- The DL command is normally activated and used by the Composer software. The command should not be used manually.
- The start memory address for downloading is defined by the LP[1] command.
- Each data payload is terminated by the 16-bit checksum for the send message.
- The DL command automatically clears the Program Ready internal flag.
- The DL command takes time to execute because it needs to burn and verify.

Attributes:		
	Type:	Command, String
	Source:	RS-232
	Restrictions:	MO=0, Program not running
	Unit modes:	All
	Activation:	Immediate

See also:
LS, CC, LP[N]

Reference chapter in the *SimplIQ Software Manual:*
Chapter 6, ”Program Development and Execution”

DV[N] - Reference Desired Value

Purpose:
Reports the reference commands to the position, speed and current controllers of the drive. DV[N] reports the final value of the controller references, as synthesized by all their sources: software reference generators, external reference inputs and external control loops.

- **For UM=1**, DV[1] reports the torque command. DV[2] and DV[3] report zero.
- **For UM=2**, DV[2] reports the speed command. DV[1] reports the torque command derived by the speed controller. The speed command may differ from the desired speed as specified by the JV command and the analog input, because the filters and the profiler "smooth the edges" before transferring the desired speed as a speed command to the speed controller. DV[3] returns zero.

  The speed command reported by DV[2] is generated by the sum of an external analog reference and a software reference. This sum is further processed for acceleration and speed limiting as well as the switch response. DV[4] and DV[5] retrieve the external and software components of DV[2].
- **For UM=3, 4, 5**, DV[1] reports the torque command derived by the speed controller, DV[3] reports the command to the position controller and DV[2] reports the speed command, which is the rate of change of DV[3]. The position command may differ from the desired position as specified by software commands and by superimposed analog input, because the stop manager may affect the position command to the controller.

  The position command reported by DV[3] is generated by the sum of an external follower reference and a software reference. This sum is further processed for acceleration speed limits, as well as the switch response. DV[6] and DV[7] retrieve the external and the software components of DV[3].
- **In summary**:
  DV[1] reports the current command value.
  DV[2] reports the velocity command value.
  DV[3] reports the position command value.
  DV[4] reports the external speed command (0 for RM=0).
  DV[5] reports the software speed command.
  DV[6] reports the external position command (0 for RM=0).
  DV[7] reports the software position command.

When the motor is disabled (MO=0), DV[N] returns zero.

Attributes:	**Type:**	Status report, Real
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Index range:	[1…7]
	Unit modes:	All

Reference chapter in the *SimplIQ Software Manual:*
Chapter 11, "Unit Modes"

EC - Error Code

Purpose:
Reports the processing status of the last accepted command that returned an error.

Notes:

- When the processing of a command fails, the error code is returned immediately with a question mark in the response to that command. The error code returned with the command response is binary, so it may not be easily seen. The EC command returns a printable (ASCII) value of the error code.
- The EC command cannot be used reliably from the Composer Smart Terminal because the Composer generates continuous communication with the servo drive. The returned EC value probably reflects the status of the latest Composer command, not the status of the last Smart Terminal command.

The follow table lists the error codes reflected by the EC command.

Error Code	Error String / Description	Example / Remedy
2	Bad command. The interpreter has not understood the command.	XF=2; is an error because there is no XF command. MC=2; is an error because the MC cannot be changed by the interpreter.
3	Bad index. Attempt to access a vectored variable out of its boundaries.	DV[6] is an error because the index range is 1 - 3. Observe the index range for the used command.
5	Has no interpreter meaning. An unrecognized character has been found where a command was expected.	A*=3 is an error because a command mnemonic consisting of two alphabetic characters was expected.
6	Program is not running.	This command requires a running program.
7	Mode cannot be started - bad initialization data.	This error is returned when preset values of a function are wrong. For example, there may be a conflict between the first index in the PVT table (PV) and the available write pointer (MP[6]) when PVT motion begins.
11	Cannot write to flash memory. An error interfacing the serial flash has occurred.	Most probably a hardware problem.

Error Code	Error String / Description	Example / Remedy
12	Command not available in this unit mode.	PA=1000 is an error if UM=2, because the position command cannot be given in this mode.
13	Cannot reset communication – UART is busy. Modification of the parameters of the serial communication has been attempted while the communication line is busy.	
18	Empty assign. The right side of an equation is missing.	AC=; is an error because the interpreter expects a numerical value to appear after the = sign.
19	Bad command format. An unresolved syntax error in the command has occurred.	Refer to this manual for the correct command syntax.
21	Operand out of range. Assignment of an illegal value to a parameter has been attempted.	JV=100,000,000 returns this error because the required speed is beyond the limits of the drive.
22	Zero division.	JV=0; PX=1000/JV returns this error.
23	Command cannot be assigned.	BG=3000 returns this error because BG is an execution command that does not have a value.
24	Bad operator. An unrecognized character has been found in an expression where an operator has been expected.	IA[1]=3$VX is an error because $ is not a recognized operator.
25	Command not valid while moving.	PV=n while in PVT motion is an error because the PV=n command sets the read pointer of the PVT table manually; this pointer is set automatically in PVT mode.
26	Motion mode not valid. A Begin Motion was attempted but the parameters of the motion were not properly set.	PV=n; BG is an error if the first valid line of the PVT table is smaller than the last valid line.

Error Code	Error String / Description	Example / Remedy
28	Out of limit range. A command was specified out of its permitted limits.	VH[2]=1000; SP=2000 is an error because the latter command specifies that point-to-point motions should reach the speed of 2000 counts/sec, whereas the first command limits the maximum speed command to 1000.
30	No program to continue. An XC command has been issued but there is no halted program to continue.	
32	Communication overrun, parity, noise or framing error.	Ensure that communication lines are well connected with adequate ground, and that the baud rate and other communication parameters are set consistently for master and slave. Also reports loss of characters in buffer when hardware storage is exceeded.
36	Bad commutation table. The data points in the back EMF table HV[N] do not form a valid back EMF function.	This error normally occurs while attempting MO=1. At MO=0, the HV[N] table can be updated, so its consistency is not required.
37	Two or more Hall sensors are defined for the same location.	One of the following: CA[4]=CA[5], CA[4]=CA[6] or CA[5]=CA[6].
39	Motion start specified for the past. The time requested for synchronized motion has elapsed.	
41	Command not supported by this product. An attempt has been made to assign an illegal value to the command.	YA[4]=3 attempts to set a type of auxiliary encoder as analog. It causes this error because SimplIQ drives do not work with analog encoders.
42	No such label. The program does not contain a label with the specified name.	XQ##FOO will return this error if neither the label ##FOO nor the function with the name FOO exists in the user program.

Error Code	Error String / Description	Example / Remedy
43	CAN state machine fault (object 0x6040 on DS402).	Reset the fault by sending the control word through CAN communication (the value of CAN object 0x6040 must be set to 0x80). Refer to the description of the 0x6040 and 0x6041 objects in the Elmo *CANopen Implementation Manual*.
45	Returned error from subroutine. Occurs when a return op-code has no valid address to return to.	
46	May not use multicapture homing mode with stop event. Occurs when trying to set multiple capture events with a STOP between events.	The following cannot be set: HM[4]=0; HM[1]=3.
47	User program does not exist. XQ or XC returns this error if a program has not been loaded to and successfully verified by the drive.	
50	Stack overflow. A CPU exception was detected. This error reflects either a hardware problem or a faulty power supply.	Use the CD command to determine the details of what has occurred. If "Called handler" is "none" and "Failure address" is non-zero, then TS is too short and there was a real-time overflow. Record the entire string of the CD command and call your service center for technical support.
53	Only for current. Command is applicable only in torque control modes UM=1 or 3.	TC=2 (Set torque to 2A) is an error in UM=2, because in this mode, the torque command is set automatically by the controller so as to achieve the desired speed.
54	Bad database. Cannot start the motor, because the setup data is not consistent.	If CA[4]==CA[5], two physical Hall sensors are defined as the same logical sensor, thereby preventing powering the motor.
55	Bad context. A command that is not applicable in the present context has been attempted.	This error is caused by privileged commands used in auto-setup sessions.

Error Code	Error String / Description	Example / Remedy
56	The product grade does not support this command.	User may have attempted to set or activate features that are available only for the Advanced SimplIQ models.
57	Motor must be off. This command cannot be used when the motor is on.	CA[25]=1 sets the order of firing the motor phases, thereby controlling the motor direction. This parameter cannot be set while the motor is on, because it will immediately destabilize the feedback loop.
58	Motor must be on. This command cannot be used when the motor is off.	PA=1000 is an error if MO=0. The absolute position reference is automatically set to the present position at MO=1, so that setting PA at MO=0 is pointless.
60	Bad unit mode. Something not supported in this unit mode has been attempted.	PT=5 is an error in UM=1 because PT motion requires position control.
61	Data base reset. The database has been restored to factory defaults after the parameters loaded from the flash memory failed a consistency check.	This error may occur after upgrading the drive version, if the newer version uses a different database structure.
64	Cannot set the index of an active table.	When the ECAM table is active, an index cannot be changed. Only a location beyond the active table limits may be changed.
65	Disabled by SW. Motion could not begin because a switch programmed to abort motion was active when MO=1 was tried.	Check the IL[N] switch definition settings and compare them to the actual switch reading (use the IP command).
66	Drive not ready. The motor could not be powered due to: ▪ Over- or under-voltage ▪ Over-temperature ▪ Short circuit (a shorted motor or a hardware problem) ▪ Hall sensor problem	Check the servo drive status (SR command or MF command)
67	Recorder is busy. A recording process is in progress and the recorder settings cannot be changed. Recorded data cannot be fetched.	Let the recorder complete its job, or use the RR=0 command to kill the recording process.

Error Code	Error String / Description	Example / Remedy
69	Recorder usage error. Something illegal was attempted with the recorder.	RC=2; RR=2 and later BH=1 is an error because an attempt is made to bring a vector that was not recorded.
70	Recorder data invalid. Cannot upload recorded data because the recorder contains no valid data.	Recorder settings (such as RC=n) have been changed since the last records were made or the recorder has not been operated at all since power up.
71	Homing is busy. Cannot change the modulo count (XM or YM) while homing is in progress.	Terminate homing processes using HM[1]=0, HY[1]=0.
72	Must be even.	(XM[2] - XM[1]) = 5 is an error because the difference is an odd number.
73	Please set position. An attempt to set the position counts modulo to a smaller number than the present position was made.	PX=2000; XM[1]=-500, XM[2]=500 is an error because the PX value is out of range.
77	Buffer too large. A string command that is too long (more than 255 characters in a single command) has been sent.	Check the command syntax.
78	Out of program range. An attempt to load a program larger than the drive storage capabilities has been made.	The amount of allocated memory for the user program is stated in the drive *User Manual*.
80	ECAM data inconsistent. The jumps between consecutive ECAM table points are greater than 32, 767 counts and therefore cannot be interpolated.	
81	In “Quick stop” mode. Occurs only if a CAN master used the DS402 standard control word to block motor movements.	In “Quick stop” mode, it is impossible to begin a software motion. Use CAN 0x6040 object to release the “Quick stop” state.
82	Program is running. Cannot load a new program, compile a program or start program execution.	Wait until the program finishes, or use the HP command or KL command to stop the program.
83	CMD not for program. An attempt has been made to use a command (such as XQ, DL, LS or DF) that has a NotProgram flag.	The next expression XQ##START; inside a user program is an error because this command has a NotProgram flag.

Error Code	Error String / Description	Example / Remedy
84	The system is not in point to point mode.	A PR (position relative) cannot be set in non-PTP mode, because it has no reference position from which to start.
90	CAN state machine is not ready (object 0x6041 on DS402).	Set the drive to the "Switched on" state machine by setting the relevant transitions to the control word, object 0x6040. Refer to the description of NMT services in the Elmo *CANopen Implementation manual*.
93	There is a wrong initiation value for this command.	Reset queue length before updating queries.
95	Too large for modulo setting.	The modulo range is inconsistent with the ER[3] value. Refer to the ER[N] command.
96	User program time out. Execution of a single user program line was more than expected (more than 3 seconds). The SimplIQ drives stops program execution.	
97	RS232 receive buffer overflow. Characters arrived through RS-232 at too high a rate, causing internal storage to exceed its capacity. No more space is left to store new characters.	
99	The auxiliary feedback entry does not configure as output during the activation of Output Compare	
100	The requested PWM value is not supported	The PWM frequency that was requested cannot be used with the drive.
105	Speed loop KP out of range. Value of KP[2] or one of KG[64]…KG[126] is out of numeric range.	
106	Position loop KP out of range. Value of KP[3] or one of KG[127]…KG[189] is out of numeric range.	

Error Code	Error String / Description	Example / Remedy
111	KV[N] vector is invalid. Invalid values in KV[N] parameters.	Refer to the "Advanced Filter" chapter in the SimplIQ *Software Manual*. If the vector was configured by the Composer auto-tuning wizard, email Technical Support for assistance.
112	KV[N] defines scheduled block but scheduling is off. Invalid values in KV[N] parameters.	See the syntax of KV[N] parameters in the "Advanced Filter" chapter of the SimplIQ *Software Manual*. If the vector was configured by the Composer auto-tuning wizard, email Technical Support for assistance.
113	Exp task queue is full. Internal error during auto-tuning process.	
114	Exp task queue is empty. Internal error during auto-tuning process.	
115	Exp output queue is full. Internal error during auto-tuning process.	
116	Exp output queue is empty. Internal error during auto-tuning process.	
117	Bad KV setting for sensor filter. Invalid setting for KV[76]…KV[87].	See KV command section of this manual.
118	Bad KV vector	This can happen when KV parameters are not set according to the correct feedback with either length or value restriction.
119	Bad Resolver Filter	When the filter KV, set for resolver feedback, is beyond its legal range.
120	Bad number of blocks for Resolver filter	When the Resolver filter contains a wrong number of blocks.
121	Resolver is not ready	When the initiation procedure of the Reslover was not completed, and the motor is being enabled.
127	Modulo range must be positive. XM[2] is less or equal to XM[1] or YM[2] is less or equal to YM[1].	

Error Code	Error String / Description	Example / Remedy
128	Bad variable index in database - internal compiler error. Index of the variable in the database is not correct.	An internal compiler error occurs due to a corrupted database. In such a case, email Technical Support for assistance. Attach the Composer date and version (found in the Help menu) and the program you attempted to compile.
129	Variable is not an array. Cannot access a scalar variable defined according to its index in square brackets as an array in the user program.	Assume that a scalar variable has been defined in the user program as a: long a; The expression a[0]=1; is wrong because a is defined as scalar and not an array.
130	Variable name does not exist. For SimplIQ internal use.	
131	Cannot record local variables. For SimplIQ drive internal use.	
132	Variable is not an array. For SimplIQ drive internal use.	
133	Mismatched number of user/system function input arguments. An attempt was made to call user/system function with the number of input arguments not as defined.	• rnd(4.6,7.7) ; This expression is wrong, since system function rnd() expects only one input argument. • Calling user function by XQ command with number of input arguments not as defined in user program.
134	Cannot run local label with XQ command.	XQ##START; when START is defined in the user program inside a user function it is consider to be a local label and therefore it is illegal to use it with the XQ command.
137	Program already compiled. An attempt was made to download a user program before previous one was erased.	Use the CP command before downloading a new user program.
139	The number of breakpoints exceeds the maximum number.	

Error Code	Error String / Description	Example / Remedy
140	An attempt to set/clear breakpoint at the non-relevant line. Internal IDE error.	For every line of the text program, there is a corresponding lien of compiled code. This error appears during an attempt to set a breakpoint at a non-corresponding line of compiled code.
141	Boot identity parameters section is not clear. Internal error during download of boot identity parameters.	
142	Checksum of data is not correct. Internal error during download of boot identity parameters.	
143	Missing boot identity parameters. Internal error during download of boot identity parameters.	
144	Numeric stack underflow. An attempt has been made to retrieve an entry from an empty stack.	
145	Numeric stack overflow. An attempt has been made to push a value to the numeric stack when it is full.	▪ User program contains very complex code requiring more stack space than is available. It may also be that there are too many called subroutines. ▪ An expression in the command line of the interpreter is too complex; it calls too many functions, so that the numeric stack has overflowed. ▪
146	Expression stack overflow. An attempt has been made to push a value to the expression stack when it is full.	An expression in the command line of the interpreter is too complex: it calls too many functions, so that the expressions tack has overflowed.
147	Executable command within math expression. An attempt has been made to assign an executable command.	BG=3; is wrong because BG is an executable command and cannot be assigned.

Error Code	Error String / Description	Example / Remedy
148	Nothing in the expression. An attempt has been made to evaluate an empty expression.	AC=; is wrong because the assign value is missing.
149	Unexpected sentence termination. An expression terminator appears in the middle of the expression.	5+3+;
150	Sentence terminator not found. The expression is too long to be evaluated (exceeding the maximum length).	Try to shorten the expression.
151	Parentheses mismatch. There is a mismatch between opening and closing parentheses. Pertains to both parentheses and brackets.	sin(2; is wrong because a closing parenthesis is absent.
152	Bad operand type. There is a mismatch between the actual value type and the expected value type.	▪ The DB command syntax is wrong (this command requires strict syntax; trying to set an unexpected floating point value causes this error) ▪ An internal compiler error has occurred due to a mismatch between operand type and its addressing mode. Contact Technical Support ▪ .
154	Address is out of data memory segment. Variable address in the data segment exceeds the data segment size.	This is internal compiler error is caused by corrupted compiled code. In such a case, email Technical Support for assistance. Attach the Composer date and version (in the Help menu) and the program you attempted to compile.
155	Beyond stack range. Compiled code contains a pointer to the stack entry, exceeding the actual stack range (STACK_IMMEDIATELY addressing method).	This internal compiler error is caused by corrupted compiled code. In such a case, email Technical Support for assistance. Attach the Composer date and version (in the Help menu) and the program you tried to compile .

Error Code	Error String / Description	Example / Remedy
156	Bad opcode. Compiled code contains mismatched addressing mode.	Internal compiler error caused by corrupted compiled code. In this case, email Technical Support for assistance. Attach Composer date and version (in Help menu) and program you tried to compile.
157	No available program stack. An attempt was made to run too many user programs simultaneously.	For future use.
158	Out of flash memory range. Failure in download and upload process: an attempt to access flash memory because its size.	Try to use IDE tools for down-loading or uploading. For more details about the internal and serial flash memory mapping.
159	Flash verify error. Failure in download and upload process: checksum does not match.	Possible hardware problem. Contact Technical Support.
160	Program aborted by another threat. Failure while running one virtual machine aborts all other virtual machines.	For future use.
161	Program is not halted. Execution of a command that requires user program to be halted.	Activation of XC command while the virtual machine is not in halted state.
162	Badly formatted number. Floating point number exceeds the valid range supported by the SimplIQ.	
164	EC command (not an error).	For internal use.
165	An attempt to access serial flash while busy. Failure on reading serial flash memory, possible due to hardware problem.	Contact Technical Support.
166	Out of modulo range.	XM[1]=-1000, XM[2]=1000 and PA=2000 is an error because the modulo range is [-1000…999]. Therefore, the position of PA=2000 can never be reached.
167	Infinite loop in for loop - zero step	k=1:0:10; causes this error.
168	Speed too large to start motor. MO=1 or motor started with Enable switch while motor was rotating too quickly.	Starting a running motor may fail if the back EMF is too high and induces an immediate excessive motor current.

Table 2-12: **Processing Error Codes**

Attributes:	**Type:**	Status report, Integer
	Source:	RS-232, CANopen
	Restrictions:	None
	Unit modes:	All

See also:
MF, SR

EF[N] - Encoder Filter Frequency

Purpose:
Filters encoder signal in order to improve its noise immunity. Because the logic of the quadrature decoder must sense transitions, the inputs are first run through a glitch filter. This filter has a digital delay line that samples four time points on the signal and verifies that a majority of the samples are at a new state before outputting the new state to the internal logic. The sample rate of this delay line is programmable, to adapt to a variety of signal bandwidths.

When an analog encoder is used, the basic signal, before interpolation, is filtered using the same method.

EF[N] sets the sample rate of the corresponding digital glitch filter: EF[1] for the main encoder and EF[2] for the auxiliary encoder. A counter increases or decreases to the value of EF[N]. When the count reaches the specified value, the counter is reset and the filter takes a new sample of the raw A, B, Index and Home input signals. If EF[N] is zero the digital filter is bypassed.

If EF[N] is large, the encoder reading noise immunity will be better, but true fast transitions (occurring by fast speed) may be dismissed as false. A number that is too small may cause the counting of noise pulses.

A good value for the required delay of the encoder filter is ¼ of the minimum time expected between transitions.

Example:
Suppose that the maximum speed of a motor is 10,000 rpm and that the motor is equipped with an encoder with 1000 lines (4000 counts/rev with resolution multiplication). The expected minimum encoder transition time is:

$$\frac{60 \text{ sec/min}}{4000 \text{ cpr} * 10{,}000 \text{ rpm}} = 1.5 \text{ µsec}$$

The minimum required encoder signal stable time should be set to about:

$$\frac{1.5 \text{ µsec}}{4} = 400 \text{ } n\text{sec}$$

The ranges of encoder frequency filtering are as follows:

EF[1] / EF[2]	**0**	**1**	**2**	**. . .**	**K**	**127**
Filter time	25 nsec	200 nsec	300 nsec	. . .	100*(K+1) nsec	12.8 µsec

When the interpolator is used, the time difference between consecutive signal changes may be shorter. This is dictated by the interpolator's specifications and not from the motor speed.

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	0 (RS), Non-volatile
	Range:	[0...127]
	Index range:	[1, 2]
	Unit modes:	All
	Activation:	Immediate

EM[N] - ECAM Parameters

Purpose:
Determines the behavior of ECAM (Electronic CAM) motions. With ECAM, the position reference to the drive is not directly proportional to the summed external inputs, but is rather a function of them. The ECAM parameters apply only to position modes (UM=3 and 5) and when the position reference is derived from the auxiliary encoder input (RM=1, FR[3]=non-zero).

Parameter	Description
EM[1]	Asserts whether the ECAM function is active: 0: Direct eternal follower referencing 1: Active linear ECAM 2: Active cyclical ECAM Set EM[1] to enter a change in the EM[2], EM[3], EM[4], EM[5] and EM[7] parameters.
EM[2]	Last valid index of the ECAM table. Maximum value is 1024.
EM[3]	Starting position: the value of the input to the ECAM function for which the output of the ECAM function will be ET[EM[5]] (ET of EM[5]).
EM[4]	Auxiliary input (ΔPY) distance between consecutive points in the ECAM table ET[N].
EM[5]	First valid index of the ECAM table.
EM[6]	Index for the next head pointer when using CAN for fast ECAM table loading.
EM[7]	Last segment shortening. Used to generate an ECAM table with an input range that is not an integer multiple of EM[4].
EM[8]	Read-only report of position in the ECAM table. When ECAM motion is not active, EM[8] reports 0.

Table 2-13: **ECAM Parameters**

Notes:

- When EM[1]=1 or EM[1]=2, the active ECAM table entries — ET[EM[5]] . . . ET[EM[2]] — cannot be changed. Other members of the ET[N] array *may* be changed.
- Parameter EM[6] takes effect immediately.
- Parameters EM[2], EM[3], EM[4], EM[5] and EM[7] are activated only when EM[1] is set. In this manner, the next work segment can be programmed while the present work segment is executing.
- Setting EM[1] to 1 or 2 will fail if EM[2] is less or equal to EM[5], or if EM[4] is less or equal to EM[7].
- Changing the last segment with EM[7] may cause a reference jump.

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	Advanced model only
	Default values:	EM[1]=0, EM[2]=2, EM[3]=0, EM[4]=1000, EM[5]=1, EM[6]=1, EM[7]=0, EM[8]=0 (RS), Non-volatile
	Range:	EM[1]: [0…2] EM[2]: [2…1024] EM[3]: Position counter range EM[4]: [1…32,000] EM[5]: [1…1023] EM[6]: [1…1024] EM[7]: 1…32,000]
	Index range:	Write: [1…7] Read: [1…8]
	Unit modes:	UM=3, 4, 5
	Activation:	See previous Notes

See also:
RM, FR[N]

Reference chapter in the *SimplIQ Software Manual:*
Chapter 12, ”The Position Reference Generator”

EO – Echo Mode

Purpose:
Sets or resets the communication echo mode, which is used for communication checks.

- EO=1 enables echo mode
- EO=0 disables it.

With RS-232 communication, the EO command sends an immediate echo character for every terminal character. The echo transmission is deferred to the command response string.

Attributes:		
	Type:	Parameter, Integer
	Source:	RS-232
	Restrictions:	None
	Default value:	1, Volatile
	Range:	[0, 1]
	Unit modes:	All
	Activation:	Immediate

ER[N] – Maximum Tracking Error

The Tracking Error is the difference between the command and its feedback.

Purpose:

- ER[2] defines the maximum allowed velocity error (abs(DV[2]-VX)) in counts/second. If the error exceeds this value, the motor is automatically disabled and the Error Limit fault is activated.
- ER[3] defines the maximum allowed position error in counts:
  for UM=5: abs(DV[3]-PX)
  for UM=4: abs(DV[3]-PY)
  If the error exceeds this value, the motor is automatically disabled and the Error Limit fault is activated.

Attributes:		
	Type:	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	ER[2]: None ER[3] - Refer to note below
	Default values:	ER[2]=400,000, ER[3]=400,000 (RS), Non-volatile
	Range:	[0…20,000,000]
	Index range:	[2, 3]
	Unit modes:	UM=2 for ER[2], UM=4, 5 for ER[3]
	Activation:	Immediate

The ER[3] value is restricted to modulo settings. The values ranges are:
For UM=5: [0..(XM[2]-XM[1])/4-1]
For UM=4: [0..(YM[2]-YM[1])/4-1]
In case of failure, an error is set during the next motor enabling (MO=1).

Typical applications:
Decrease ER[N] as much as possible in order to use it as a protection mechanism in case of control failure, as when the feedback signal is lost.

See also:
MF, MO, SR

Reference chapter in the *SimplIQ Software Manual:*
Chapter 14, ”Limits, Protections, Faults and Diagnosis”

ET[N] - Entries for ECAM Table

Purpose:
In the ECAM process, the position reference is set to a tabulated function, called the ECAM function, of the external inputs. The ET[N] vector stores the tabulated values of the ECAM function.

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None (see Notes below)
	Default values:	0 (RS), Non-volatile
	Range:	$[(-2^{30}+1)\ldots(2^{30}-1)]$
	Index range:	[1…1024]
	Unit modes:	UM=3, 4, 5
	Activation:	Immediate

Notes:

- When the motor is enabled (MO=1) and the ECAM table is running (EM[1]=1), you may manipulate entries that are not in an active part of the table. This provides an on-the-fly programming of the next motion. Refer to the EM[N] command for active table descriptions.
- With CAN communication, you may program the ECAM table with a fast Auto-increment mode.
- When the ECAM table is not used, ET[N] can be used as a general-purpose non-volatile memory.

See also:
EM[N], UM, RM

Reference chapter in the *SimplIQ Software Manual:*
Chapter 12, ”The Position Reference Generator”

FF[N] - Feed Forward

Purpose:
Defines how much of the position reference derivative is fed as reference to the speed controller.

- For most UM=5 applications, FF[2]=1.
- For most UM=4 applications, FF[2] is the number of counts traveled by the main (speed) feedback, while the position (auxiliary) feedback travels one count.

The FF[1] parameter defines the factor with which the second derivative of the position reference is injected to the torque controller as a direct torque command.

Attributes:	**Type:**	Parameter, Real
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	FF[1]=0 (RS), Non-volatile
		FF[2]=1 (RS), Non-volatile
	Range:	[0.0…4,194,303.5]
	Index range:	[1, 2]
	Unit modes:	UM=4, 5
	Activation:	Immediate

Examples:
Suppose that a gear motor with a reduction ratio of 5 drives a load. The motor has an encoder with 1000 lines. The motor speed is used for the inner feedback loop. The load position, measured by an encoder with 2000 lines, is used as feedback for the outer loop.

To prevent a steady-state error at constant speed, set: $FF[2] = \frac{2000}{1000 * 5} = 0.4$.

See also:
UM

Reference chapter in the *SimplIQ Software Manual:*
Chapter 11, ”Unit Modes”

FR[N] - Follower Ratio

Purpose:

- FR[1] defines the follower ratio for current (UM=1).
- FR[2] defines the follower ratio for velocity (UM=2).
- FR[3] defines the follower ratio for position (UM=3, 5).

When UM=1, the auxiliary reference is composed of the analog input and external PWM signals. The FR[1] parameter scales the ratio between the Duty Cycle of the PWM signal and the reference to the current loop (UM=1, RM=1). FR[1] may be changed on-the-fly at any time.

When UM=2, the auxiliary reference is composed of the analog input and the auxiliary feedback readout or external PWM signal.

FR[2] can be used as a follower ratio of the master motor's quadrature encoder or as a follower ratio of the PWM Duty.

When UM=3 or UM=5, the auxiliary reference is composed of the auxiliary feedback readout. The FR[3] parameter scales the ratio between the auxiliary feedback position and the reference to the position loop (UM=3, 4, 5, RM=1, EM[1]=0), or the input to the ECAM table (UM=3, 4, 5, RM=1, EM[1]=1, 2). FR[3] may be changed on-the-fly at any time. For EM[1]=0 (follower) and EM[1]=2 (cyclical ECAM), changing FR[3] does not imply an abrupt change to the external position controller reference. For EM[1]=1 (linear ECAM), changing FR[3] *does* imply an abrupt change in the external position controller reference. FR[3] can be modified while the motor is enabled if the software reference is not active (MS<1). In such cases the software position reference is corrected to avoid a possible motor jump.

Attributes:		
	Type:	Parameter, Real
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=0: none MO=1: RM=1 and MS<2
	Default value:	1 (RS), Non-volatile
	Range:	[-32,000…32,000]
	Index range:	[1, 2, 3]
	Unit modes:	UM=1, 2, 3, 5
	Activation:	Immediate

See also:
PY, RM, YM[N], YA[N], PW[N]

Reference chapter in the *SimplIQ Software Manual:*
Chapter 12, "The Position Reference Generator"

GS[N] - Gain Scheduling

Purpose:
Defines the gain scheduling process.

SimplIQ drives are scheduled according to the controller and the command states. This may be necessary due to either the difference between the low-speed behavior and the high-speed behavior of the plant or because the inertia changes with position dependence. The process of assessing the situation and varying the controller parameters online accordingly is called "gain scheduling."

The following table lists the gain scheduling parameters. Unused indices are reserved for compatibility with older drives.

Parameter	Description	Values
GS[0]	No encoder count over which speed loop is opened	0…500
GS[1]	Minimum speed command for speed and dual gain scheduling (counts/sec)	0…16*62*256 internal speed units (1 speed unit = counts/Ts/2^16)
GS[2]	Use scheduled gains: 0: No 64: Yes	0, 64
GS[4]	Upward gain of gain scheduling filter	0…32,767
GS[5]	Downward gain of gain scheduling filter	0…32,767
GS[9]	NL factor for position controller	0…60,000,000
GS[10]	Position error coefficient for position gain scheduling to raise gains	0…1,200
GS[14]	Maximum speed error for which KP[N] is cleared in speed controller if no encounter count exceeds GS[0]	$0…2^{31}-1$
GS[15]	Gain scheduling step: 0: 256 1: 128 N: $256/2^N$	0…3

Table 2-14: **Gain Scheduling Parameters**

The GS[N] array is normally programmed by the Composer IDE. Manipulate it only if you are sure of what you are doing.

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=0
	Default values:	GS[0]=500, Non-volatile GS[4]=30,000, Non-volatile GS[5]=30,000, Non-volatile GS[9]=60,000,000, Non-volatile GS14]=32,000, Non-volatile Other defaults=0, Non-volatile
	Range:	According to previous table
	Unit modes:	UM=2, 4, 5
	Activation:	Immediate

HL[N] - Over-speed Limit and Position Range Limit

Refer to LL[N] - Low Feedback Limit.

HM[N] - Homing, Capture and Flag

Purpose:
Sets the parameters of the main homing and capture process, by which the drive sets a trap for a user-defined event. When the event occurs, the *SimplIQ* can:

- Modify a position counter (homing)
- Log the exact position of the event (capture)
- Flag a digital output (flag)

An event is a change in a digital input signal. The polarity of the change is defined by the IL[N] command. Values in HM[3] are duplicated for compatibility reasons.

HM[N] (Index)	Value	Description
1 Activation mode	0	Disarm homing process. HM[1] is automatically reset to 0 when homing is complete.
	1	Arm homing process. The sequence is activated according to the last HM[2] to HM[6] values. HM[7] and HM[8] are cleared.
2 Absolute value		Value to load, according to method of HM[5]. Absolute value is limited to position counter range.
3 Event definition	0	Immediate: Trigger is the receipt of HM[1]=1.
	1/2	Event according to Main Home switch (capture).
	3	High transition[1] of Index pulse (capture).
	4	Low transition[2] of Index pulse (capture).
	5/6	Event according to defined FLS switch.
	7/8	Event according to defined RLS switch.
	9/10	Event according to defined DIN1 switch.
	11/12	Event according to defined DIN2 switch.
	13/14	Event according to defined DIN3 switch.
	15/16	Event according to defined DIN4 switch.
	17/18	Event according to defined DIN5 switch.
	19/20	Event according to defined DIN6 switch.
	21/22	Event according to defined DIN7 switch.
	23/24	Event according to defined DIN8 switch.
	25/26	Event according to defined DIN9 switch.
	27/28	Event according to defined DIN10 switch.

[1] Index input level changes from low to high.
[2] Index input level changes from high to low.

HM[N] (Index)	Value	Description
4 After-event behavior. Defined as the time in which HM[1] decreases to 0.	0	In UM=2, 3, 4, 5: Stop immediately using SD deceleration value. In torque mode (UM=1), do nothing.
	1	Set digital output, equivalent to OP=HM[6].
	2	Do nothing.
5 What to set for PX during event	0	Absolute setting of position counter: PX=HM[2].
	1	Relative setting of position counter: PX=PX (at event) -HM[2]
	2	Do nothing.
6 Output value		Digital output value if HM[4]=1. Only outputs defined as general output are affected.
7 PX captured value		The captured value of PX (read only). The position value is captured before PX is changed according to HM[5].
8 PY captured value		The captured value of PY (read only). The position value is captured at the next controller sampling time and therefore may be inaccurate to (4*VX*TS*10^{-6}) counts.

Table 2-15: **HM[N] Command Values**

Notes:

- Elmo drives have a different number of digital inputs. The value of HM[3] may differ according to that. Please consult the drive's *Installation Guide* for details.
- HM[2] - HM[6] can be changed during the home search procedure. The activation of the parameters is considered upon reception of the next HM[1]=1 (or higher).
- If HM[2] is set to a value beyond XM[1] and XM[2], the actual main position will not be updated when homing is complete.
- Each homing event is attached to a predefined functionality (FLS, General Purpose, Home and so on). If the corresponding input is not defined first, the homing procedure may never end. Refer to the IL[N] command.
- The homing and capture procedures can be carried out in any unit mode (UM=1, 2, 3, 4, 5) and reference mode (RM=0, 1).
- In external reference mode (RM=1), when HM[4]=0, the software portion of the reference is stopped, while the external portion is not. In such cases, the motor continues to move according to the analog reference.
- Homing can be safely carried out in PTP and jog position motion modes. With PT and PVT modes, the online reloading of the position counter can lead to an immediate, automatic MO=0 due to excessive position error.

- When the Index or Home signal captures PX, the PY captured value is taken at the next position controller sampling time (4 TS period). It may differ from the PY value at the capture time by up to $4*TS*10^{-6}*VY$ counts.
- When the Index or Home signal captures PX, the digital output of HM[6] is set only at the next position controller sampling time (4 TS period).

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	0, Volatile
	Range:	HM[1]: [0, 1] HM[2]: According to PX command HM[3]: [0…24] (see the first note, above) HM[4]: [0…2] HM[5]: [0…2] HM[6]: According to OP command HM[7 - 8]:Read only, according to PX, PY
	Index range:	[1…8]
	Unit modes:	All
	Activation:	Immediate

See also:
HY[N], XM[N], IL[N], OL[N], PX, PY, EF[N], IF[N], SD

Reference chapter in the *SimplIQ Software Manual:*
Chapter 13, "Sensors, I/O and Events"

HP – Halt Program Execution

Purpose:
Stops the execution of the user program and the automatic routines. The HP command freezes the status of the program and does not reset it. A later XC command will resume the program from the instruction at which the program was halted. Pending interrupts will remain pending.

An HP command issued when no program is running does nothing and sets no error code.

Attributes:	**Type:**	Command, No value
	Source:	RS-232, CANopen
	Restrictions:	None
	Unit modes:	All
	Activation:	Immediate

The HP command, together with a later XC command, may be used if a communicated command sequence must be executed consecutively, without program interference.

See also:
KL, XQ, XC

HV[N] - Commutation Tables

Purpose:
For internal use only.

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=0, RR=-1, Program not running
	Range:	[-32,767…32,767]
	Index range:	[0…2303]
	Unit modes:	All
	Activation:	Upon setting the correct checksum at HV[2303]

Reference chapter in the *SimplIQ Software Manual:*
Chapter 9, "Commutation"

HX - Hexadecimal Mode

Purpose:
Sets or resets the hexadecimal mode for reporting integer parameter values.

- With HX=0, integers are reported as decimal numbers.
- With HX=1, integers are reported as hexadecimal numbers.

HX does not affect floating-point reports.

The HX parameter allows easy reading of the digital inputs (IP), servo drive status (SR), motor faults (MF), recorder settings (RC) and other variables that have the bit-field attribute.

The HX parameter is not required for setting values. The commands BH=1024 and BH=0x400 are equivalent, as 0x400 equals its decimal equivalent 1024.

Attributes:		
	Type:	Parameter, Integer
	Source:	RS-232
	Restrictions:	None
	Default value:	0, Volatile
	Range:	[0, 1]
	Unit modes:	All
	Activation:	Immediate

HY[N] - Auxiliary Homing, Capture and Flag

Purpose:
Sets the parameters of the auxiliary homing and capture process, by which the *SimplIQ* sets a trap for a user-defined event. When the event occurs, the *SimplIQ* can:

- Modify the auxiliary position counter (homing)
- Log the exact position of the event (capture)
- Flag a digital output (flag)

An event is a change in a digital input signal. The polarity of the change is defined by the IL[N] command. Values in HY[3] are duplicated for compatibility reasons.

HY[N] (Index)	Value	Description
1 Activation mode	0	Disarm homing process. HY[1] is automatically reset to 0 when homing is complete.
	1	Arm homing process. The sequence is activated according to the last HY[2] to HY 6] values. HY[7] and HY[8] are cleared.
2 Absolute value		Value to load, according to method of HY[5]. Absolute value is limited to position counter range.
3 Event definition	0	Immediate: Trigger is the receipt of HY[1]=1.
	1/2	Event according to Auxiliary Home switch (capture).
	3	High transition of Index pulse (capture).
	4	Low transition of Index pulse (capture).
	5/6	Event according to defined FLS switch.
	7/8	Event according to defined RLS switch.
	9/10	Event according to defined DIN1 switch.
	11/12	Event according to defined DIN2 switch.
	13/14	Event according to defined DIN3 switch.
	15/16	Event according to defined DIN4 switch.
	17/18	Event according to defined DIN5 switch.
	19/20	Event according to defined DIN6 switch.
	21/22	Event according to defined DIN7switch.
	23/24	Event according to defined DIN8 switch.
	25/26	Event according to defined DIN9 switch.
	27/28	Event according to defined DIN10 switch.
4 After-event behavior. Defined as the time in which HY[1] decreases to 0.	0	In UM=2, 3, 4, 5: stop immediately, using SD deceleration value. In torque mode (UM=1), do nothing.
	1	Set digital output, equivalent to OP=HY[6].
	2	Do nothing.

HY[N] (Index)	Value	Description
5 What to set for PY during event	0	Absolute setting of position counter: PY=HY[2].
	1	Relative setting of position counter: PY = PY (at event) -HY[2]
	2	Do nothing.
6 Output value		Digital output value if HY[4]=1. Only outputs defined as general output are affected.
7 PY captured value		Captured value of PY, before any modification by the HY[N] command (read only).
8 PX captured value		Captured value of PX (read only).

Table 2-16: **HY[N] Command Values**

Notes:

- HY[2] - HY[6] can be changed during the home search procedure. The activation of the parameters is considered upon reception of the next HY[1]=1 (or higher).
- If HY[2] is set to a value beyond YM[1] and YM[2], the actual auxiliary position will not be updated when homing is complete.
- Each homing event is attached to a predefined functionality (FLS, General Purpose, Home and so on). If the corresponding input is not defined first, the homing procedure may never end. Refer to the IL[N] command.
- The homing and capture procedures can be carried out in any unit mode (UM=1, 2, 3, 4, 5) and reference mode (RM=0, 1).
- In external reference mode (RM=1), when HY[4]=0, the software portion of the reference is stopped, while the external portion is not. In such cases, the motor continues to move according to the analog reference.
- Homing can be safely carried out in the PTP and jog position motion modes. With PT and PVT modes, the online reloading of the position counter can lead to an immediate, automatic MO=0 due to excessive position error.
- When the Index or Home signal captures PY, the PX captured value is taken at the next position controller sampling time (4 TS period). It may differ from the PX value at the capture time by up to $4*TS*10^{-6}*VY$ counts.
- When the Index or Home signal captures PY, the digital output of HY[6] is set only at the next position controller sampling time (4 TS period).

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	0, Volatile
	Range:	HY[1]: [0, 1] HY[2]: According to set PY command HY[3]: [0…24] HY[4]: [0…2] HY[5]: [0…2] HY[6]: According to OP command HY[7 - 8]: Read only according to PX, PY
	Index range:	[1…8]
	Unit modes:	All
	Activation:	Immediate

See also:
HX, XM[N], IL[N], OL[N], PX, PY, EF[N], IF[N]

Reference chapter in the *SimplIQ Software Manual:*
Chapter 13, "Sensors, I/O and Events"

IB[N] - Input Bits Array

Purpose:
Provides read access to digital input bits.

IB[N] reports the status of the corresponding input bits, according to the definition in IP. If IB[N] is "1", the corresponding Nth bit in IP is logically active.

Use the IB[N] command to reference general purpose inputs, limit switches and other indications (such as Stop, Begin or Enable) individually.

IB[N] may be more convenient than IP for program decisions and branching. However, it is not appropriate for the synchronized reading of several input bits. If a synchronized reading of several digital inputs is desired, use the IP command.

Refer to the IP command for more details about the role of each bit.

Attributes:		
	Type:	Status report, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Index range:	[0…31]
	Unit modes:	All

IB[N] reports according to the polarity programmed for the relevant digital input by the IL[N] command.

See also:
IP, IL[N]

ID, IQ - Read Active Current and Reactive Current

Purpose:
Gets the active and the reactive components of the motor current, in amperes.

A brushless motor carries alternating currents in its phases. The alternating currents in the motor phases create a rotating magnetic field, which can be projected in two directions. The first magnetic field component is aligned with the magnetic direction of the rotor; it produces no mechanical torque. The other magnetic field component is perpendicular to the magnetic direction of the rotor and produces all the mechanical torque.

IQ[Amp] is the component of the motor phase current that creates effective torque. The current controller attempts to make IQ equal to the current command. ID is the component of the motor phase current that does *not* create torque. The current controller tries to null ID.

When the motor is off (MO=0), IQ and ID are not calculated and return zero.

Attributes:	**Type:**	Status report, Real
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Unit modes:	All

See also:
AN[N], MC, PL[N], CL[N]

Reference chapter in the *SimplIQ Software Manual:*
Chapter 10, ”The Current Controller”

IF[N] - Digital Input Filter

Purpose:
Filters the drive digital inputs in order to overcome switch bounding. IF[N] defines a time period in milliseconds. Input pulses of shorter duration than IF[N] are rejected. Pulses longer than IF[N] in milliseconds are sensed.

Each index entry [1 - 10] refers to a digital input [1 - 10] respectively. The input filtering is accomplished by the software. For this reason, in order to ensure that an input pulse is sensed, its length must be IF[N]+2*TS, where TS is the sampling time.

Example:
If the speed sampling time is 210 microseconds and IF[1]=1, the minimum stable time for an input change to be sensed is 1050 microseconds (210*5). A pulse must be at least 1 millisecond+2*210 microseconds =1.42 millisecond long in order to ensure that it is captured by the digital input.

Attributes:	**Type:**	Parameter, Float
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	0 (RS), Non-volatile
	Range:	[0…1000] msec
	Index range:	[1…10] … refer to the first note
	Unit modes:	All
	Activation:	Immediate

Notes:

- Elmo drives supports a different number of digital inputs. Accordingly, the index range may differ between drives. Please consult the drive's *Installation Guide* for information about its digital inputs.
- If input 5 or 6 is used as Home input, the corresponding parameter — IF[5] or IF[6]— does not apply. Parameter EF[1] or EF[2] is used instead.
- The IF[N] commands can also be applied in simulation mode.

IL[N] – Input Logic

Purpose:
Defines the logic level and functional behavior of the digital inputs. The drive has several non-committed digital inputs. Each of these inputs can be programmed to a specific function and logic level. In addition, the IL[N] function enables the simulation of a digital input. This option is convenient for testing and debugging user programs.

A digital input serves only one dedicated function, which can be reflected in the following commands/features:

- User program auto routine (#@AUTO_##)
- Homing procedure (HM[N], HY[N])
- IP command
- IB[N] command

Notes:

- The response to a digital input is made only according to the definition of IL[N]. For example, if digital input #2 is defined by IL[2] as RLS (Reverse Limit switch), changes in the connector pin of digital input #1 will not be reflected in IB[1] commands. IB[1] will read continuously zero. In this case, IB[11] will reflect this input status.
- When a digital input is activated, the relevant bit in IP/IB[N] is set. Refer to the IP and IB[N] commands for more details.
- Inputs 5 and 6 also serve as high-speed home/capture inputs, used independently by the HM[N] and HY[N] commands. The logic level for the home inputs is also defined by the IL[N] command.
- If inputs 5 or 6 are used as home inputs, the corresponding parameters IF[5] and IF[6] do not apply. The parameters EF[1] and EF[2] are used instead.

The following table summarizes the functions that may be attached to each digital input pin. The function details are given in the next table. The prefix “Hard” indicates that the function applies to the stop manager, not to the motion reference generator. The term “Soft” indicates that the function applies to the motion reference generator. The term “Hard and Soft” indicates that the function applies both to the stop manager and to the position reference generator.

IL[N] Bits	Meaning	Values
0	Logic levels	0: Low level active. The function attached to this switch is activated when *no* current flows through the input opto-coupler.
		1: High level active. The function attached to this switch is activated when current flows through the input opto-coupler.
1 - 4	Function behaviors (next table)	0: Inhibit (INH); shut servo driver, freewheel. For RM=1 and UM=1, the *SimplIQ drive* will retry starting the motor automatically when the inhibit function is released.
		1: Stop immediately under control; hard stop only.
		2: Ignore.
		3: General purpose.
		4: Hard-enable forward direction only (RLS).
		5: Hard-enable reverse direction only (FLS).
		6: Begin.
		7: Stop immediately under control; soft stop only.
		8: Home switch for IL[5] only.
		9: Auxiliary Home switch for IL[6] only.
		10: Simultaneous activation of the hard and soft stop functions (functions 1 and 7).
		11 - Abort.
		12 - 15: Reserved.
5	Simulation mode	0: Read value from digital input pin. 1: Read value from bit 6, regardless of pin state.
6	Simulation value	Value to set for pin if bit 5 is on.
7 - 15	Reserved	

Table 2-17: **IL[N] Functions**

Command Value	Active Level	When Active . . .
IL[N]=0	Low	Shut servo drive, freewheel.
IL[N]=1	High	Shut servo drive, freewheel **Note:** It is high recommended *not* to use this state. The motor may spin when the input wire is cut or disconnected.
IL[N]=2	Low	Stop immediately under control: soft and auxiliary stop.
IL[N]=3	High	Stop immediately under control: soft and auxiliary stop.
IL[N]=4	Low	No function is attached. Ignore the switch.
IL[N]=5	High	No function is attached. Ignore the switch.
IL[N]=6	Low	General purpose.
IL[N]=7	High	General purpose.
IL[N]=8	Low	Hard-enable forward direction only (RLS).
IL[N]=9	High	Hard-enable forward direction only (RLS).
IL[N]=10	Low	Hard-enable reverse direction only (FLS).
IL[N]=11	High	Hard-enable reverse direction only (FLS).
IL[N]=12	Low	Begin: activates BG command.
IL[N]=13	High	Begin: activates BG command.
IL[N]=14	Low	Stop immediately under control: soft stop only. Activates the ST command.
IL[N]=15	High	Stop immediately under control: soft stop only. Activates the ST command.
IL[5]=16	Low	Enable the Main Home sequence.
IL[5]=17	High	Enable the Main Home sequence.
IL[6]=18	Low	Enable the Auxiliary Home sequence.
IL[6]=19	High	Enable the Auxiliary Home sequence.
IL[N]=20	Low	Stop immediately under control: stop both software trajectory and auxiliary reference.
IL[N]=21	High	Stop immediately under control: stop both software trajectory and auxiliary reference.
IL[N]=22	Low	Abort motion. Shut servo drive, freewheel.
IL[N]=23	High	Abort motion. Shut servo drive, freewheel.

Table 2-18: **Possible Values for IL[N]**

Function 0: Inhibit (freewheel)
Servo is off (MO=0). The motor is *not* under control. No current is applied through the motor phases. If the motor was previously running, it will continue to coast on its own inertia. The motor fault code (see the MF command) is 0x10. If the unit mode is UM=1 (torque control) or UM=2 (velocity control) and an external command is active (RM=1), a motor restart will be attempted when the switch is "not active." This attempt is made within a few (no less than 10) milliseconds. In addition, when restarting the motor the #@AUTO_ENA automatic routine can be activated.

Function 1: Hard stop immediately under control
The function behavior depends on the unit mode:

UM	Action
Torque (UM=1)	Set torque command to zero.
Speed (UM=2)	Set speed command to zero immediately at the deceleration of the SD parameter.
Position (UM=3, 4, 5)	Slow down to complete stop using the deceleration of the SD parameter.

Function 2: Input is ignored
This serves no function in the system and always reads zero in the IP/IB[N] indications.

Function 3: General purpose (GPI)
No special function. Serves as an uncommitted input. The input may be used in the user program and homing sequences as simple digital input. In addition, general purpose inputs can activate ##AUTO_DIN automatic routines in the user program.

Function 4: Hard-reverse limit switch
The function activates the ##AUTO_RLS routine in the user program. In addition, it has the following unit mode dependent actions:

UM	Action
Torque (UM=1)	Allow only positive torque commands. Negative torque demands yield zero motor current.
Speed (UM=2)	Allow only positive speed command (external or internal). If, at the time of switch sensing, the speed command was negative, the speed command will converge to zero using the stop deceleration (SD).
Position (UM=3, 4, 5)	Allow only positive position command increments (external and internal). If, at the time of switch sensing, the speed was negative, the position command will decelerate to complete stop using the deceleration of the SD parameter.

Function 5: Hard-forward limit switch
The function activates the ##AUTO_FLS routine in the user program. In addition, it has the following unit mode dependent actions.

UM	Action
Torque (UM=1)	Allow only negative torque commands. Positive torque demands yield zero motor current.
Speed (UM=2)	Allow only negative speed command (internal or external). If, at the time of switch sensing, the total speed command was positive, the speed command will converge to zero using the stop deceleration (SD).
Position (UM=3, 4, 5)	Allow only negative position command increments (external and internal). If, at the time of switch sensing, the speed was positive, the position command will decelerate to a complete stop using the deceleration of the SD parameter.

Function 6: Begin
The function behaves like a software BG command, activating the ##AUTO_BG routine in the user program. In addition, it has the following unit mode dependent actions:

UM	Action
Torque (UM=1)	Nothing.
Speed (UM=2)	Set software speed command to JV.
Position (UM=3, 4, 5)	Set software position command to the activated motion mode (PA, JV, PT, PV).

Function 7: Software Stop
The function behaves like a software ST command, activating the ##AUTO_ST routine in the user program. In addition, it has the following unit mode dependent actions:

UM	Action
Torque (UM=1)	Nothing.
Speed (UM=2)	Reduce the software speed command to zero, using the deceleration SD.
Position (UM=3, 4, 5)	Set software position command to complete stop, using the deceleration SD.

Function 8: Main Home switch
This function activates the ##AUTO_HM routine in the user program. When the function is selected, digital input connector pin #5 serves as the Home/Capture switch for the feedback defined as main. Only IL[5] can be programmed to this function. Refer to the HM[N] command for more information.

Function 9: Auxiliary Home switch
This function activates the ##AUTO_HY routine in the user program. When the function is selected, digital input connector pin #6 serves as the Home/Capture switch for the feedback defined as auxiliary. Only IL[6] can be programmed to this function. Refer to the HY[N] command for more information.

Function 10: Hard and soft stop
The function activates the ##AUTO_ST routine in the user program. It stops the motor under control, stopping the response to external reference and applying the software ST command simultaneously. This function actually activates function 1 and function 7 simultaneously.

UM	Action
Torque (UM=1)	Set the torque command to zero.
Speed (UM=2)	Reduce the software speed command to zero, using the stop deceleration SD. Reduce the controller speed command to zero, using the deceleration SD.
Position (UM=3, 4, 5)	Set the software position command to a complete stop, using the stop deceleration SD. Bring the controller reference command to a complete stop, using the deceleration SD.

Function 11: Abort motion

The behavior is similar to the Inhibit function with the exception that the "Abort" input release will not start the motor automatically. After the Abort is activated, MO=1 must be set either by communication or by the internal User Program.

The function activates the #@AUTO_ER routine, if it exists, in the user program.

Notes:

- Make sure that the drive you use has the actual digital input that is programmed. Failing to do so will not generate an error. Not all drives have the same digital input entries. Nevertheless no error indication would be given in case a "none existing" digital input is programmed. For example the Harmonica drive has 6 physical digital inputs. An attempt to set IL[10]=8 would not generate an error but the RLS function would be programmed to the drive
- Use the Inhibit freewheel function with care. When the drive is shut, the motor applies no torque. Turning off a drive might leave the motor spinning until it stops by friction. In some situations, this may be dangerous.
- When a switch is released, the attached function terminates. Functions 2, 3 and 4 (Full Stop, RLS and FLS) do not change the drive reference command. When the switch is released, the reference command (speed or position) is recovered. In order to ensure that reference recoveries do not generate discontinuities, the SD, VL[2] and VH[2] limits are used.
- IP and IB[N] can be used to detect a logically active switch of all defined functions, excluding function 2 ("No function is attached").

Attributes:	**Type**:	Parameter, Bit-field
	Assignment:	Yes
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default value:	IL[1]=0
		IL[2…10]=7 (RS), Non-volatile
	Range:	According to previous description
	Index range:	[1…10]
	Unit modes:	All
	Activation:	Immediate

Reference chapter in the *SimplIQ Software Manual:*
Chapter 12, " The Position Reference Generator"

See also:
UM, RM, JV, PX, BG, IP, IB[N], HM[N], HY[N]

IP - Input Port

Purpose:
Reports an active or non-active state of a digital input. A digital input is considered to be *active* when the associated function is logically active. The functionality and logic levels are defined in the IL[N] command.

IP logic is always positive. When the digital input is *active*, the relevant IP bit is set.

The report is a bit-field, defined in the following table:

Bit	Description	Associated Function in IL[N] Command
0	General purpose input 1 is active	3
1	General purpose input 2 is active	3
2	General purpose input 3 is active	3
3	General purpose input 4 is active	3
4	General purpose input 5 is active	3
5	General purpose input 6 is active	3
6	Main home switch	8
7	Auxiliary home switch	9
8	Soft stop	7, 10
9	Hard stop	1, 10
10	Forward Limit (FLS)	5
11	Reverse Limit (RLS)	4
12	INH (Enable) switch	0
13	Hardware BG	6
14	Abort function	11
15	Not used; always 0	
16	Digital input 1 logical pin state	
17	Digital input 2 logical pin state	
18	Digital input 3 logical pin state	
19	Digital input 4 logical pin state	
20	Digital input 5 logical pin state	
21	Digital input 6 logical pin state	
22	Digital input 7 logical pin state	
23	Digital input 8 logical pin state	
24	Digital input 9 logical pin state	
25	Digital input 10 logical pin state	
26 - 31	Reserved; always 0	

Table 2-19: **IP - Input Port**

Notes:

- Each type of Elmo drive supports a different number of digital inputs. Please consult the drive's *Installation Guide* for more information about its inputs.
- For compatibility reason inputs 7-10 do not have an indication for the "General Purpose" function and cannot be used for user program AUTO routine as well. Bits 22-25 will still be set regardless to the above.
- The logical state of digital input pins 1 to 10 — as indicated in bits 16 to 25 — is reflected in the logic level required in the relevant IL[1] to [10], respectively. IB[N] may be more convenient than IP for user program decisions and branching. However, it is not recommended for the synchronized reading of several input bits. If such a reading is needed, use the IP command.

Attributes:	**Type:**	Status report, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Unit modes:	All

See also:
IB[N], IL[N]

JV- Jogging Velocity

Purpose:
Sets the motor speed. In speed control mode (UM=2), the JV parameter specifies the software speed command. In un-profiled mode (PM=0), the speed command is set to JV immediately. In profiled mode (PM=1), the speed command is gradually changed to JV, according to the AC, DC and SF parameters.

In the position control modes (UM=4, 5), the JV setting defines a constant speed software command. The value of JV defines the speed of the motion.

The parameters AC, DC and SF determine the acceleration limits for reaching final speed. In position-jogging mode (JV), and if the position feedback sensor is set to modulo counting (refer to XM[N] and YM[N]), a position-controlled motor can rotate forever. The position reading will jump each modulo count according to the last modulo setting, but the speed will remain steady.

Notes:

- Jog mode is recommended for homing procedures, because it does not require information about starting position or destination.
- In position mode (UM=4, 5), a jogging command is under position control. The JV parameter determines the rate at which the position command changes.
- In stepper mode (UM=3), JV determines the rate at which the electric angle command changes.
- In position control mode (UM=3, 4, 5), JV not only sets the speed of motion but it also states that the next motion will be a constant speed jog.
- For all relevant modes (UM=2, 4, 5), the value of JV must be set between VL[2] and VH[2]. Setting JV out of this range will invoke an "Out of limit" error code.

Attributes:		
	Type:	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	UM=2: None UM=3, 4, 5: MO=1
	Default values:	0 (RS), Non-volatile
	Range:	Velocity range
	Unit modes:	UM=2, 3, 4, 5
	Activation:	BG

See also:
AC, BG, DC, PX, XM[N], YM[N], SF, SP, VH[N], VL[N], MS

Reference chapter in the *SimplIQ Software Manual:*
Chapter 12, "The Position Reference Generator"

KG[N] - Gain Scheduled Controller Parameters

Purpose:
Specifies the parameters of the gain scheduled speed or position controller. The KG[N] parameters apply only if the controller gains are scheduled (GS[2]=64).

The following table details the use of the KV[N] parameters array:

Index	KG[N] Value	Length
0	Reserved	1
[1…63]	KI for inner loop	63
[64…126]	KP for inner loop	63
[127…189]	KP for outer loop	63
[190…252]	GSIndexTable table	63
[253…315]	Parameter 1 for scheduled advanced filter	63
[316…378]	Parameter 2 for scheduled advanced filter	63
[379…441]	Parameter 3 for scheduled advanced filter	63
[442…504]	Parameter 4 for scheduled advanced filter	63

Table 2-20: **KG[N] Gain Scheduled Controller Parameters**

Attributes:	**Type:**	Parameter, [1…189]: Real; [190…504]: Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	RS=0
	Range:	See section 15.4 "The Gain Scheduling Algorithm" in the *SimplIQ Software Manual*
	Index range:	[0…504]
	Activation:	Immediate

Reference chapter in the *SimplIQ Software Manual:*
Chapter 15, "The Controller"

KI[N], KP[N] - PI Parameters

Purpose:

- KI[1], KP[1] defines the PI current control filter.
- KI[2], KP[2] defines the PI velocity control filter.
- KP[3] defines the gain of the position controller.

The parameters KP[2], KI[2] and KP[3] apply only if the controller gains are fixed (gain scheduling is not used: GS[2]=0).

Attributes:	**Type:**	Parameter, Real
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	RS=0
	Range:	KI[N]>0 KP[N]>0
	Index range:	[1…3]

See also:
KV[N], GS[N]

Reference chapters in The *SimplIQ Software Manual:*
Chapter 10, "The Current Controller;" Chapter 15, "The Controller"

KL - Kill Motion and Program

Purpose:
Halts program execution and stops the motor. The KL command stops the execution of the user program threads and automatic routines. It also issues the MO=0 motor disable command. KL freezes the status of the program and does not reset it. A later XC command will resume the program from the instruction at which the program was halted. Pending interrupts will remain pending.

A KL command issued when no program is running does nothing, and sets no error code.

Attributes:	**Type:**	Command, No value
	Source:	RS-232, CANopen
	Restrictions:	None
	Unit modes:	All
	Activation:	Immediate

If the motor is on when KL is used, using XC to continue the program may fail, because the interrupted program expects the motor to be on, and it may use commands that are restricted to the MO=1 state.

See also:
HP, XQ, XC

KV[N] - High-order Controller Filter Parameters

Purpose:
Specifies the parameters of the following filters:

Filter	Parameters	Maximum Order
Speed controller high-order filter	KV[0]…KV[47]	
Position controller high-order filter	KV[48]…KV[75]	
Analog position sensor filter	KV[76]…KV[87]	Order 4 (2 blocks)
Analog reference to speed controller	KV[88]…KV[99]	Order 4 (2 blocks)

Table 2-21: **KV[N] Filter Partitions**

The KV[0] parameter defines whether or not the speed controller high-order filter is used:

- If KV[0]=0, the high-order filter is not used.
- If KV[0]=100, the high-order filter is defined by the rest of the KV[N] parameters.

Similarly, KV[48]=0 deactivates the position controller high-order filter, KV[76]=0 deactivates the analog position sensor filter, and KV[88]=0 deactivates the filtering of the analog reference to the speed controller.

The KV[N] array specifies almost arbitrary linear filters. To learn how the KV[N] parameters specify a filter and to see a programming example, refer to the "Filters" chapter of the The *SimplIQ Software Manual.*

Attributes:

Type:	Parameter, Integer
Source:	Program, RS-232, CANopen
Restrictions:	MO=0
Default values:	RS=0
Range:	Previous text
Index range:	[0…99]
Activation:	MO=1

See also:
KI[N], KP[N], KG[N], GS[N], UM

Reference chapter in The *SimplIQ Software Manual:*
Chapter 14, "Filters"

LC - Current Limit Flag

Purpose:
Reports the status of the current limiting process. You may select two different current limit specifications: The peak limit PL[1] specifies how much current can be applied to the motor for short time periods (PL[2]) and the continuous limit CL[1] specifies how much current can be applied to the motor continuously.

LC returns values according to the following table:

Value	Description
0	The motor current is limited to the limit PL[1], or the motor is off.
1	The motor current is limited to the continuous limit CL[1].

Attributes:

Type:	Status report, Integer
Scope:	Program, RS-232, CANopen
Restrictions:	None
Unit modes:	All

See also:
MC, PL[N], CL[N]

Reference chapter in The *SimplIQ Software Manual:*
Chapter 10, "The Current Controller"

LD - Load Parameters from Flash

Purpose:
Loads all non-volatile variables from the flash memory to the RAM and resets all volatile variables to their default values.

Before accepting the loaded parameters, LD tests them as follows:

- The variables written in the flash memory can be read. The variables *cannot* be read if the flash memory is brand new and no parameters have ever been saved in it, or after a major firmware update.
- The variables in the flash memory are all in their permitted range. This test should not fail, as the legality of all variables is tested prior to saving.

If any of these tests fail, the contents of the flash are ignored and all non-volatile variables are set to their factory default (RS) states.

Certain exceptional variables are not reset by the LD command. The communication (PP[N]) parameters are retrieved from the flash memory, but are not set into action. The parameter PP[1] retains its old values in order to ensure communication continuity. The newly-loaded RS-232 communication parameters may be activated by PP[1]=1, and the new CAN parameters are activated by a "Communication Reset" NMT message.

The LD command may take a few milliseconds to perform, because it completely recalculates the drive database. At this time, the communication routines are disabled. If an LD command is executed by an RS-232 command, a CAN message may be lost in the meanwhile, and vice versa.

Attributes:	**Type:**	Command, No value
	Source:	RS-232, CANopen
	Restrictions:	MO=0, Program not running
	Unit modes:	All
	Activation:	Immediate

If an LD command fails, CD will report the reason for the failure by adding the string "Couldn't load from serial flash" followed by the reason for the failure.

The save() function used within a User Program does not check the integrity of the data before saving it to the flash memory.

See also:
SV, RS, CD

LL[N] - Low Feedback Limit

Purpose:

- *Speed limits*
  The parameters LL[2] and HL[2] define the limits of the allowed motor speed. If the motor speed exceeds HL[2] or is lower than LL[2], the drive is automatically disabled and the "Speed High Limit" fault (MF=0x20,000) is activated Speed limits are restricted by the TS value according to the following:
  For *analog encoder, resolver, tachometer, potentiometer and digital halls:*
  SpeedLimit = minimum between 80,000,000 and 8e9/TS.
  For quadrature encoder:
  SpeedLimit = minimum between 20,000,000 and 8e9/TS.
- *Position limits*
  LL[3] and HL[3] define the allowed motor position range for UM=3, 4 and 5. If the motor position is smaller than LL[3] or larger than HL[3], the motor is automatically disabled and the "Position High Limit" fault (MF=0x400000) is activated. In order to re-enable the motor, modify the current position (PX) within the ranges of HL[3] and LL[3].

Attributes:		
	Type:	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=0 and For LL[2] and HL[2]: HL[2]>LL[2] For LL[3] and HL[3]: HL[3] >LL[3]
	Default values:	LL[2]=-1,000,000 HL[2]=1,000,000 LL[3]=-15,000,000,000 HL[3]=15,000,000,000 (RS), Non-volatile
	Range:	LL[2]: Velocity range HL[2]: Velocity range $-2^{31} \leq LL[3] \leq 2^{31} - 1$ $-2^{31} \leq HL[3] \leq 2^{31} - 1$
	Index range:	[2, 3]
	Unit modes:	HL[2], LL[2]: UM=2, 3, 4, 5 HL[3], LL[3]: UM=3, 4, 5
	Activation:	Immediate

Note: The position counter is subject to a modulo count; refer to position counter range.

See also:
VL[N], VH[N], MF, SR, MO, XM[N]

Reference chapter in The *SimplIQ Software Manual:*
Chapter 14, "Limits, Protections, Faults and Diagnosis"

LP[N] - List Properties

Purpose:
Sets the properties of the serial flash data upload and download by the next LS and DL commands.

- LP[1] sets the start byte address of the next LS transmission, or the byte address for starting the storage of the next DL transmission.
- LP[2] sets the size — in bytes — to be transmitted at the next LS.
- LP[3] returns the start byte address of the user program (read only).
- LP[4] returns the size of the user program storage (read only).

Attributes:	**Type:**	Parameter, Integer
	Scope:	RS-232, CANopen
	Restrictions:	None
	Default values:	LP[1] and LP[2] set to 0 on power on
	Range:	LP[1]: [0…262,144] LP[2]: [0…247]
	Index range:	[1…4]
	Unit modes:	All
	Activation:	Immediate

See also:
CC, DL, LS

Reference chapter in The *SimplIQ Software Manual:*
Chapter 6, "Program Development and Execution"

LS - List User Program

Purpose:
Uploads data from the serial flash to the host, according to the parameters of LP[N]. The most common use of LS is to retrieve the user program and to retrieve the personality (firmware partition) data of the drive.

LS begins to send data from the byte address of LP[1] in the serial flash memory. The length of the transmitted data is LP[2] bytes.

The format of the LS message is:
[data payload][esc][checksum]
where:

- The data payload is in the hex-binary format.
- esc is the 0x1b (27 decimal) character.
- The checksum is calculated for 16 bits in 2's complements.

Use the LS command with care because it is not entirely "safe": while the program listing uploads to the communication lines, no program instructions are executed, and no communicated commands are interpreted.

Attributes:	**Type:**	Command, No value
	Source:	RS-232
	Restrictions:	None
	Unit modes:	All
	Activation:	Immediate

See also:
CC, DL, LP[N]

Reference chapter in the *SimplIQ Software Manual:*
Chapter 6, "Program Development and Execution"

MC – Maximum Peak Driver Current

Purpose:
Reports the maximum phase current allowed for the drive, in amperes. This command informs the software about the type of servo drive used with the controller.

Attributes:	**Type:**	Report, Real
	Source:	Program, RS-232, CANopen
	Restrictions:	Read only
	Unit modes:	All

You may limit the current for a specific application using the PL[1] and CL[1] commands.

See also:
IQ, ID, CL[N], PL[N]

MF – Motor Failure

Purpose:
Reports the reason why the motor has been automatically shut down (set to MO=0). MF normally reports zero (as default). The fact that the motor has been automatically shut down is reflected as a bit in the status register (SR) report, and MF provides the detailed information.

After a fault, the MF value remains fixed, even if the reason for the fault no longer exists. MF is automatically set to zero on the next motor enable MO=1.

Fatal faults (CPU exceptions) are permanent and can be resent only be power reset.

The following table lists the type of faults reported by the MF value.

Reported Fault	Value	Bit
The motor is on, or the last motor shutdown was the normal result of a software command.	0	
1. Resolver feedback is not ready – Resolver angle was not found yet. 2. Analog encoder or Resolver feedback is either lost or with too low amplitude.	0x1	0
Reserved.	0x1	0
Reserved.	0x2	1
Feedback loss: no match between encoder and Hall location. Available in encoder + Hall feedback systems.	0x4	2
The peak current has been exceeded. Possible reasons are drive malfunction or bad tuning of the current controller.	0x8	3
Inhibit.	0x10	4
Reserved.	0x20	5
Two digital Hall sensors were changed at the same time. Error occurs because digital Hall sensors must be changed one at a time.	0x40	6
Speed tracking error DV[2] - VX (for UM=2 or UM=4, 5) exceeded speed error limit ER[2]. This may occur due to: ▪ Bad tuning of the speed controller ▪ Too tight a speed error tolerance ▪ Inability of motor to accelerate to the required speed due to too low a line voltage or not a powerful enough motor	0x80	7
Position tracking error DV[3] - PX (UM=5) or DV[3] - PY (UM=4) exceeded position error limit ER[3]. This may occur due to: ▪ Bad tuning of the position or speed controller ▪ Too tight a position error tolerance	0x100	8

Reported Fault	Value	Bit
▪ Abnormal motor load, or reaching a mechanical limit		
Cannot start because of inconsistent database. The type of database inconsistency is reflected in the status SR report, and in the CD CPU dump report.	0x200	9
Too large a difference in ECAM table.	0x400	10
Heartbeat failure. Error occurs only if drive is set to abort under heartbeat failure in a CANopen network (object 0x6007 in CAN object dictionary is set to 2).	0x800	11
Servo drive fault. Error described according to the servo drive fault detail bits 13 - 15 in the MF report. Refer to following table.	0x1000	12
Servo drive fault detail bit 1. Refer to following table.	0x2000	13
Servo drive fault detail bit 2. Refer to following table.	0x4000	14
Servo drive fault detail bit 3. Refer to following table.	0x8000	15
Failed to find the electrical zero of the motor in an attempt to start it with an incremental encoder and no digital Hall sensors. The reason may be that the applied motor current did not suffice for moving the motor from its position.	0x10,000	16
Speed limit exceeded: VX<LL[2] or VX>HL[2].	0x20,000	17
Stack overflow - fatal exception. This may occur if the CPU was subject to a load that it could not handle. Such a situation can arise only due to a software bug in the drive. Use the CD command to get the CPU dump and report to your service center.	0x40,000	18
CPU exception - fatal exception. Something such as an attempt to divide in zero or another fatal firmware error has occurred. Use the CD command to get the CPU dump and report to your service center.	0x80,000	19
Reserved.	0x100,000	20
Motor stuck - the motor is powered but is not moving according to the definition of CL[2] and CL[3].	0x200,000	21
Position limit exceeded: PX<LL[3] or PX>HL[3] (UM=5), or PY<LL[3] or PY>HL[3] (UM=4).	0x400,000	22
Reserved.	0x10,000,000	28
Cannot start motor.	0x20,000,000	29
Reserved.	0x80,000,000	31

Table 2-22: **Reasons for Automatic Motor Shutdown**

0x8000	0x4000	0x2000	Meaning
0	0	0	OK.
0	0	1	Under voltage. The power supply is shut down or it has too high an output impedance.
0	1	0	Over voltage. The voltage of the power supply is too high, or the servo drive did not succeed in absorbing the kinetic energy while braking a load. A shunt resistor may be required.
0	1	1	Reserved.
1	0	0	Reserved.
1	0	1	Short circuit. The motor or its wiring may be defective, or the drive is faulty.
1	1	0	Temperature. Drive overheating. The environment is too hot, or lacks heat removal. There may be a large thermal resistance between the drive and its mounting.
1	1	1	Reserved.

Table 2-23: **Bits 13 - 15 of MF Report**

Attributes:	**Type:**	Status report, Integer, Bit-field
	Scope:	Program, RS-232, CANopen
	Restrictions:	None
	Unit modes:	All

Example:
The MF report of 0x3000 indicates that the motor has been shut down due to under voltage. The under-voltage condition did not necessarily exist at the time the MF was reported. It may possibly be the result of high power consumption and a high output impedance of the power supply. In such a case, when the motor was shut down, the power consumption stopped and the power supply returned to its normal voltage.

See also:
SR, MO, TS, LL[N], HL[N], CD, PE, VE

Reference chapter in the *SimplIQ Software Manual:*
Chapter 10, "The Current Controller"

MI - Mask Interrupt

Purpose:
Selects which interrupts (automatic routines) are active.

A user program may include a main code and some automatic routines. When the program runs, the conditions for calling these routines are checked continuously. If the conditions for running a automatic routine[3] are met, it is called. At certain times, you may want to block some of the automatic routines. For example:

- An #@AUTO_RLS automatic routine may be deactivated in a homing process.
- You may want a certain code sequence to be un-interruptible.
- Certain auto-routines may be needed only when starting the program from certain labels.

MI is a bit field. Each bit in MI masks its corresponding automatic routine, preventing its execution. The MI bits are detailed in the following table:

MI Value	Masked Interrupt	Relevant Routine
1 (0x1)	Not used	0
2 (0x2)	Abort	AUTO_ER
4 (0x4)	Soft stop	AUTO_STOP
8 (0x8)	Soft begin	AUTO_BG
16 (0x10)	RLS	AUTO_RLS
32 (0x20)	FLS	AUTO_FLS
64 (0x40)	Switch enable	AUTO_ENA
128 (0x80)	Digital input 1	AUTO_I1
256 (0x100)	Digital input 2	AUTO_I2
512 (0x200)	Digital input 3	AUTO_I3
1024 (0x400)	Digital input 4	AUTO_I4
2048 (0x800)	Digital input 5	AUTO_I5
4096 (0x1000)	Digital input 6	AUTO_I6
8192 (0x2000)	Main Home event	AUTO_HM
16,384 (0x4000)	Auxiliary Home event	AUTO_HY
32,768 (0x8000)	User program error	AUTO_PERR

Table 2-24: **MI Bits**

[3] These conditions include the requirements that: no interrupt request of high priority is pending (in which case the corresponding automatic routine enters the pending list) and that the interrupt is not masked (in which case the interrupt is ignored).

Notes:

- MI is not affected by the XQ command. You should set MI to the desired value in the first lines of your user program in order to ensure that the correct automatic routines can run.
- Several interrupts may be blocked by the MI command. MI=65,535 or MI=0xffff will block all interrupts of the table. MI=48 will block only the AUTO_FLS and the AUTO_RLS interrupts. MI=MI|2 will block AUTO_ER and leave all other interrupt mask bits as is.
- If AUTO_PERR is activated, all other interrupts will be masked (MI=0x7fff).

Attributes:		
	Type:	Parameter, Integer, Bit-field
	Scope:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	0, Volatile
	Range:	[0…65,535]
	Unit modes:	All
	Activation:	Immediate

See also:
XQ, XC

Reference chapter in the *SimpIQ Software Manual:*
Chapter 5, " The *SimpIQ* User Programming Language"

MO - Motor Enable/Disable

Purpose:
Enables and disables (freewheels) the motor power.

- *Disabling the motor*
  MO=0 disables the motor. This is the idle state of the drive. The power stage is disabled and no current flow sin the motor. In this mode, the servo drive can perform various tasks that are impossible when the motor is on:
  - Boot on power up
  - Calculate and check the integrity of the drive database
  - Download new firmware and user programs
  - Save parameters in the flash memory
  - Modify setup data that cannot be modified on-the-fly, such as commutation parameters (CA[N]) and unit mode (UM)

  The servo driver is automatically disabled when a motor fault is captured (MF). An attempt to enable the motor may fail if the conditions of the fault still exist.

  If the brake function is activated, the MO=0 command duration is extended according to the BP[N] command specification.

- *Enabling the motor*
  MO=1 is the operative state of the servo drive, driving the motor and activating and executing the programmed motion. The software runs a set of tests to ensure that all conditions for running the motor are met.

  If MO is set to 1 and the motor is already on, nothing happens.

  When the motor is enabled, the drive reinitializes the internal parameters and motion drivers. The drive may fail to start if the setup data is found to be inconsistent (for example, CA[4]=CA[5]). In this case, CD commands indicate the reason for the failure.

  The last captured motor fault (MF) is reset to zero.

  In the position control modes (UM=4, 5), the motor is always started so that it does not jump. The total position control command — which consists of the internal position command and the external position command — is set to the actual present position of the motor in order to prevent the motor from jumping.

Notes:

- When RM=1 the servo drive will attempt to start (set MO=1) automatically after power on. A servo drive that has been shut down for any reason will attempt an automatic restart every few milliseconds. Automatic restart will occur only if the Enable switch (INH) is active (refer to the IL[N] command). In all other modes, the MO=0 state is maintained until MO is explicitly changed by a software command.

- In encoder-only systems (in which no digital Hall sensors are present), the commutation is calculated only once after power up upon the first MO=1 command. The motor moves a few encoder counts during the automatic commutation search.
- When the brake function is enabled, the dedicated output is released after the duration defined in BP[N]. During this time, all motion reference commands are ignored.
- After a motor fault (MF>0), the motor can be re-enabled only when the cause of the fault is no longer present.
- If MO=1 fails with the "Amplifier not ready" error code (EC=66), the exact reason for the fault can be found by querying the MF command.
- MO=0 disables the motor for 200 sampling time (about 10 milliseconds) until a new MO=1 can restart the motor.
- In the "Ready to switch on", "Switch on disabled" and "Fault" states (refer to the Elmo *CANopen Implementation Manual*), MO=1 is blocked and returns an error. These states can be command controlled only by a CAN master using the DS402 standard control word (object 0x6040).
- Setting MO=1 in the "Switched on" CAN state transfer the state of the state machine to "Operation enabled." Setting MO=0 in "Operation Enabled" or "Quick stop active" state will transfer the state machine to "Switched on." (Refer to CAN object 0x6040 in the Elmo *CANopen Implementation Manual*).

Attributes:		
	Type:	Command/Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	0, Volatile
	Range:	[0…1]
	Unit modes:	All
	Activation:	Immediate

See also:
MF, SR, CD, BP[N]

MP[N] - Motion (PT/PVT) Parameters

Purpose:
Programs the parameters of PVT or PT motions. PT or PVT motion is programmed as a sequence of points that are visited at programmed times.

The QP[N] array stores the position reference points used for PT or PVT motion. The QV[N] and QT[N] arrays store the speed and timing data that is additionally required for PVT motions. The entries of the QP[N], QV[N] and QT[N] arrays used for a certain motion may be programmed, in order to enable part of an array to be used for running the motion while another part of the array is programmed for subsequent motions.

The QP[N], QV[N] and Q[N]T arrays may be referred to as cyclical buffers. In cyclical mode, periodic motions can be set to run forever. In addition, the host may program the table on-the-fly, generating an infinite, online updated motion.

The MP[N] array defines how the QP[N], QV[N] and QT[N] tables are used for PVT and PT motions, as detailed in the following table.

Parameter	Description
MP[1]	First index in the QP[N], QV[N] and QT[N] arrays to be used for the motion.
MP[2]	Last index in the QP[N], QV[N] and QT[N] arrays to be used for the motion.
MP[3]	0: The motion is to terminate after the last (MP[2]) element of the QP[N], so QV[N] and QT[N] arrays are used. A PT or a PVT motion terminates by decelerating the motor to a complete stop, using the SD deceleration. When using CAN, an emergency object is transmitted. 1: The motion is to be cyclical, so after using the last (MP[2]) element of the QP[N], QV[N] and QT[N] arrays, the first (MP[1]) element is used again. 2: Reserved 3: CAN is being used and the motion is to be terminated without an emergency object.
MP[4]	Used for PT motions only. The number of drive sampling times between consecutive specifications of position reference points. Note that MP[4] counts sampling times for the position controller. This sampling time is given by WS[28].
MP[5]	If CANopen communication is used, allows the drive to send an emergency object to the host when only a predefined number of valid reference points has been left in the QP[N], QV[N] and QT[N] motion arrays. This way, the host is released from polling the status continuously for the motion queue. MP[5] programs the number of valid motion points remaining when an emergency object is sent to the host. If MP[5]=0, no emergency object is sent.
MP[6]	Reports the next entry index (write pointer) for the following point in the PVT/PT table. This report is required for running PVT and PT motions using CANopen and special high-speed motion referencing methods that are available only for CANopen.

Table 2-25: **MP[N] Parameters**

MP[1] and MP[2] cannot be changed during a PT or PVT motion.

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	**PVT:** $1 \leq MP[1] < MP[2] \leq 64$ **PT:** $1 \leq MP[1] < MP[2] \leq 1024$ **General:** $MP[2]-MP[1] \geq 1$
	Default values:	MP[1]=1, MP[2]=64, MP[3]=1, MP[4]=4, MP[5]=50, MP[6]=1, Volatile
	Range:	**PVT:** MP[1]: [1…64] MP[2] : [1…64] MP[3] : [0…3] MP[4] : [1…256] MP[5] : [0…62] MP[6] : [1…64] **PT:** MP[1] : [1…1024] MP[2] : [1…1024] MP[3] : [0…3] MP[4] : [1…256] MP[5] : [0…62] MP[6] : [1…1024]
	Index range:	[1…6]
	Unit modes:	UM=3, 4, 5
	Activation:	Immediate

See also:
PT, PV

Reference chapter in The SimplIQ *Software Manual:*
Chapter 12, ”The Position Reference Generator”

MS - Motion Status

Purpose:
Reports the status of the motion profiling process. MS can be used for detecting the end of motions: a PTP motion that has reached its target, or a completed PT or PVT motion.

- For **position control modes** (UM=3, 4, 5), MS reports as follows:

Value	Description
0	Motor position stabilized. The *feedback* position is steady within the ranges defined by TR. Note that value 0 is applicable only when there is a defined position target, as in PTP.
1	The *reference* for the position controller is stationary, or the motor is off (MO=0).
2	The *reference* to the position controller is dynamically controlled by one of the optional motion profilers: PTP, Jog, PT or PVT.
3	Reserved.

MS reflects only the software motions. It does not indicate anything about the behavior of the external reference command.

- For **speed control modes** (UM=2), MS reports as follows:

Value	Description
0	Not applicable for this mode.
1	▪ The *reference* to the speed controller equals the speed target. (This is always the case if no profiler mode is used: PM=0.) ▪ The motor is off (MO=0).
2	The software *reference* to the speed controller differs from the speed target. In software profiled reference mode (PM=1, RM=0), this is the case while accelerating, decelerating or smoothing to target speed JV.
3	Reserved.

Attributes:

Type:	Status report, Integer
Scope:	Program, RS-232, CANopen
Restrictions:	None
Unit modes:	All

See also:
PM, RM, DV[N], AC, DC, TR

OB[N] - Output Bits Array

Purpose:
Sets and resets an output bit. The OB[N] command only sets a digital output that is defined by OL[N] as a general purpose output.

- OB[N] for N=1…6 returns the value of digital output (N), if digital output N is defined as general purpose output. Otherwise, it returns 0.
- OB[10…15] are reserved.
- If at least one of the digital outputs is mapped to the Amplifier OK function, OB[7] returns 1 if the drive is ready (MO=1, or MO=0 and no exception such as under-voltage prevents MO=1), or 0 if the drive is not ready. If none of the digital outputs is mapped to the Amplifier OK function, OB[7] returns 0.
- If at least one of the digital outputs is mapped to the Brake function, OB[8] returns 1 if the brake is engaged, or 0 if the brake is released. If none of the digital outputs is mapped to the Brake function, OB[8] returns 0.
- If at least one of the digital outputs is mapped to the Motor enable/disable function, OB[9] returns 1 if the MO=1, or 0 if the MO=0. If none of the digital outputs is mapped to the Motor enable/disable function, OB[9] returns 0.
- For unused bits (N=[10…15]), OB[N] returns 0.

Notes:

- Each of Elmo's SimplIQ drives supports a different number of digital outputs. The number of digital outputs is specified in the drive's *Installation Guide*. The value of OB[N] varies according to the logic state of the output even if the output does not exists.
- The OB[N] syntax may be more convenient than OP for setting individual outputs. However, it is not appropriate for the synchronized setting of several output bits.
- OB[N] will not affect digital outputs defined as AOK and/or Brake. If OB[N] is applied to an output that is defined for a specific function, no error indication will be sent, but the command will do nothing.
- OB[N] sets and reads the logical values of the outputs. The voltage values at the digital output connector pins are active high or active low with respect to OB[N], according to the OL[N] setting.
- At boot, all OB[N] values not defined for the Amplifier OK function are set so that the output opto-couplers do not conduct. This way, no transition will occur at the digital outputs when the drive boots. If another value is required immediately after boot, use an #AUTOEXEC routine.
- Since Output Compare is "hard wired" to digital output #1, any setting or reading of digital output #1 will be false when Output Compare is active (because the pins are shared and Output Compare signals take precedence).

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default value:	0 (RS), Volatile
	Range:	[0, 1]
	Index range:	[1, 6] (Set); [1…15] (Get)
	Unit modes:	All
	Activation:	Immediate

See also:
OP, OL[N], BP[N]

OC[N] – Output Compare

Purpose:
Output a signal when the present position is compared to a user defined position.

The OC command generates a train of pulses according to the encoder position values.

The OC[1] command operates in two modes:

1. OC[1]=1: Absolute position mode. The first pulse is generated by the initialized absolute position defined by OC[2] (i.e. PX=OC[2] or PY=OC[2] depending on the OC[6] setting) and continues at position intervals specified by the OC[3] value. That is, PX or PY =OC[2]+k*OC[3], where k is the number of successful compared occurrences (k=1,2,...).
2. OC[1]=2: Immediate mode. The first pulse is generated immediately after the command is received by the drive (no initialized absolute position value is required) and continues at position intervals specified by OC[3] counts.

The number of occurrences in which the pulses are generated can be limited by OC[5]. The duration of a pulse can be set by OC[4].

Index	Description
1	0: Disable Output Compare 1: Accept last changes in OC[N] and enable Output Compare beginning at Absolute Position OC[2]. Note: to get proper execution of the command, the following condition should be met: OC[3]*(OC[2]-PX)>0 when the source is the main feedback, or OC[3]*(OC[2]-PY)>0 when the source is the auxiliary feedback. 2: Accept last changes in OC[N] and enable Output Compare immediately
2	The absolute position for first pulse (PX or PY value). Applicable only in absolute position mode. This value cannot exceed the modulo limit in the same direction of motion i.e. |OC[2]|-|PX| or |OC[2]|-|PY| must be positive.
3	The position interval between subsequent pulses (in encoder counts). The positive/negative value of OC[3] should be set according to the direction of the encoder motion. When the direction is positive (increasing PX value) OC[3] should be positive; otherwise it should be negative.
4	N: Pulse duration in 25*(N+1) [nSec].
5	N: Number of pulses to generate 0: Infinite Output Compare mode (train of pulses will end only with the OC[1]=0 command)
6	Output Compare Source Signal 0: Output Compare on Main feedback. 1: Output Compare on Auxiliary feedback

OC[1] returns the following values:

-1: No more pulses are being generated because the number of pulses specified in OC[5] has been reached.

0: Output Compare function is disabled.

1: Output Compare function has started but absolute position has not yet been reached; therefore, the train of pulses has not begun.

2: The train of pulses is being generated now.

Notes:

1. When performing Output Compare on Main Feedback (except incremental encoder) or on the Auxiliary Feedback, the Auxiliary Feedback entry should be configured as output (YA[4]=4) before Output Compare is initialized. Activation this function without setting the Auxiliary Feedback as output causes a "Bad auxiliary sensor configuration" error.

2. NEVER LET the frequency of Output Compare pulses become greater than 20 kHz while working on EMULATED FEEDBACK. The feedback is emulated with all feedbacks modes excluding quadrature encoder or digital halls. Output Compare on Emulated Encoders works on interrupts and uses software resources, so there is limited Output Compare signal frequency. If it becomes too high, there will be a problem with amplifier operation without any indication of error or fault.

   Nonetheless, Output Compare on quadrature Incremental Encoder Feedback (not emulation) is a hardware based feature and has no such limitation.

3. Be aware that long duration pulses cause a number of pulses to connect and produce one overlapping pulse. The drive gives no warning on such occasions.

4. When going in the direction opposite the specified direction with the same OC[3] value, the compare will occur every 65,535 - OC[3] counts.

5. When Output Compare is active, Digital Input #1 is used to produce the train of pulses. In such a case, any configuration of this output (set by OL[1], OB[1] or OP) is ignored.

6. In Output Compare at Absolute Position mode (OC[1]=1), the first produced pulse is longer (OC[4]+50nsec) than other pulses.

7. When working in Absolute Position mode (OC[1]=1), the absolute position which specified by OC[2] must be at least 10uSec away from the point in which the feature was activated .

8. The Main or Auxiliary Home capture does not function when Output Compare works on the same Main or Auxiliary feedback source.

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	OC[1]…OC[2], OC[5]…OC[6]=0 , OC[3]=100(RS), volatile OC[4]=200(RS), volatile
	Range:	OC[1]: [0…2] OC[2]: [-10^9 … 10^9] OC[3]: [-65,535 …65,535], excluding 0,-1,1 OC[4]: [1…65,535] OC[5]: [0…65,535] OC[6]: [0..1]
	Index range:	[1…6]
	Unit modes:	All
	Activation:	After OC[1]=1 or OC[1]=2

See also: YA

OL[N] – Output Logic

Purpose:
Defines the logic level and function behavior of the digital outputs. The drive has several non-committed digital outputs (2 in the Harmonica and Bassoon, 5 in the Cello, 6 in the Cornet). Each of these outputs can be programmed to a specific function and logic level.

Each OL[N] entry is dedicated to a certain output: OL[1] for the DOUT1 connector pin and OL[2] for the DOUT2 connector pin, … OL[N] for the N connector pin

The following table summarizes the functions that can be attached to the digital output pin.

OL[N] Bits	Meaning	Values
0	Logic levels	0: Low level active.
		1: High level active.
1 - 4		0: General purpose. Serves as a simple digital output function, subject to the OB[N] or OP setting.
		1: AOK, drive ready for use.
		2: Brake. Responds according to definition in BP[N].
		3: Motor enable/disable indication
		4: Reserved.
5 - 15	Reserved	

Table 2-26: **OL[N] Functions**

Terms:

- Logic level low:
  When the function is active, the opto-coupler is conducting.
- Logic level high:
  When the function is active, the opto-coupler is open circuit.
- AOK:
  Indicates that the drive is ready. The AOK signals that the physical conditions needed to run the motor are available. If AOK is flagged, the motor DC voltage is within range, the drive temperature is good, and no over-current or short-circuit has been captured during the previous 10 milliseconds. When programming this option, OB[N] and OP have no influence on the relevant output.
- Brake:
  The output is mapped to brake functionality as defined in BP[N]. When programming this option, OB[N] and OP have no influence on the relevant output. The brake function works only in positive logic.
- Motor Enable/disable

  The output is mapped to this function as defined in the MO command. When programming this option, OB[N] and OP have no influence on the relevant output.

The possible values of OL[N] are outlined in the following table.

Command Value	Active Level	When Active . . .
OL[N]=0	Low	Output serves as general purpose.
OL[N]=1	High	Output serves as general purpose.
OL[N]=2	Low	AOK: drive ready for use.
OL[N]=3	High	AOK: drive ready for use.
OL[N]=4	Low	Brake feature is active.
OL[N]=5	High	Reserved
OL[N]=6	Low	Motor enable/disable indication
OL[N]=7	High	Motor enable/disable indication

Table 2-27: **Possible Values for OL[N]**

Notes:

- The number of outputs in each type of *SimplIQ* drive differs. Nevertheless there will be no error indication when setting an output that does not physically exists in the drive. For example the Harmonica drive will receive the command OL[6]=3 with no error. As output #6 does not exist in the Harmonica no AOK indication can be retrieved from the drive.
- Since the Output Compare output is "hard wired" to physical digital output #1, any configurations of this output (OL[1]) are ignored when the feature is active.

Attributes:

Type:	Parameter, Integer, Bit-field
Process:	Yes
Assignment:	Yes
Scope:	Program, RS-232, CANopen
Restrictions:	None
Default values:	0, Non-volatile
Range:	According to basic logic table
Index range:	[1, 6]
Unit modes:	All
Activation:	Immediate

See also:
OB[N], OP, BP[N], MO, OC

OP – Output Port

Purpose:
Sets values for all uncommitted digital outputs, defined as *general purpose* by the OL[N] command. OP does *not* affect the digital output pins otherwise defined. The bits of OP[N] can be individually accessed by OB[N]. For more information, refer to the OB[N] and OL[N] commands.

When OP is queried, the status of the "Amplifier OK" , "Brake" and "Motor enable/disable" functions are reflected in bits 7 ,8 and 9, respectively.

Notes:

- The number of outputs in each type of *SimplIQ* drive differs. Accordingly, the outcome of this command may differ between drives. Please consult the drive's *Installation Guide* for more details about the amount of its digital.
- When any of the uncommitted digital outputs are defined as *general purpose*, the physical state of the output depends on the previous OP command setting and its logic level defined by OL command.
- The OB[N] syntax may be more convenient than OP for setting individual outputs. However, it is not appropriate for the synchronized setting of several output bits. If a synchronized setting of several digital outputs is desired, use the OP command.
- Since Output Compare "hard-wired" to physical digital output #1, any setting or reading of this output will be false when the feature is active

Attributes:	**Type:**	Parameter - Status, Integer, Bit-field
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	0, Volatile
	Range:	[0…63] for write For querying, as described above
	Unit modes:	All
	Activation:	Immediate

See also:
OB[N], OL[N], BP[N], OC

PA - Absolute Position

Purpose:
Specifies that the next software position command will be a PTP (point-to-point) and defines the target position for the next PTP motion. The position reference to the drive is composed of an "internal" software command and an external command, calculated from the analog inputs and the auxiliary feedback input.

In PTP motion, the drive calculates the software position reference so that a desired target position will be reached as fast as possible, subject to the speed (SP) and the acceleration limits (AC, DC, SD) (with possible smoothing: SF). A PTP command can be given any time, at any speed, regardless of the present executing motion. The drive calculates the acceleration, speed and minimum time path to the target position, starting from the present position and speed.

A PTP motion, like any other software motion, is initiated by a BG command. For example, a PA=1000 command specifies that the next BG command will activate a PTP motion, overriding any previous motion specification, and that the target position will be 1000. The new PA value causes PR to become 0. In order to make sequential PTP movements of the same size, use the PR command. Subsequent PR settings may cause the PA value to be changed.

Notes:

- The PA value must be within the modulo range: (XM[1]…XM[2] - 1) for UM=3, 5 or (YM[1]…YM[2] - 1) for UM4, and restricted by the VL[3] and VH[3] settings.
- Modulo calculation is always active; therefore, PTP motions are planned using the shorter way. For example, if XM[1]=-500, XM[2]=500, UM=5, the present position command is 490 and the next PA is set to -490, the PTP motion will be planned in the positive direction, and the resulting motion length will be 20 counts.
- A PTP motion will continue until completion even if the position counter value has been reset on-the-fly (refer to the HM[N] / HY[N][4] commands). However, the position target does not change so that, in the case of PA=n, the BG sequence may result in a different motion length than initially programmed.

Attributes:		
	Type:	Command/Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=1
	Range:	Position counter range
	Unit modes:	UM=3, 4, 5
	Activation:	BG

See also:
PR, BG, MO, XM[N], YM[N], MS, VH[N], VL[N], SP, AC, DC, SD, SF, HL[N], LL[N]

Reference chapter in the *SimplIQ Software Manual:*
Chapter 12, "The Position Reference Generator"

[4] In dual loop mode (UM=4), position feedback is PY. It can be manipulated on-the-fly by HY.

PE - Position Error

Purpose:
Returns the present position tracking error, in counts.

- In *main* feedback position mode (UM=5), PE reads:
  PE = DV[3] – PX.
  PE is read modulo-XM[N], taken the shorter way.
  For example, if XM[1]=-500, XM[2]=500, DV[3]=400 and PX=-400, PE will read 200.
- In *auxiliary* feedback position mode (UM=4), PE reads:
  PE = DV[3] – PY.
  PE is read modulo-YM[N], taken the shorter way.
  For example, if YM[1]=-500, YM[2]=500, DV[3]=400 and PY=-400, PE will read 200.

If the absolute value of PE exceeds ER[3], motion is aborted and the motion fault code MF=256 (0x100) is set. If MO=0, or if the position controller is not used (UM=1, 2 or 3), PE returns 0.

Attributes:		
	Type:	Status report, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Unit modes:	4, 5

See also:
XM[N], YM[N], ER[N], MF, UM

PK - Peek Memory

Purpose:
Returns the DSP memory dump for an address range.

Syntax:
PK=N
where N is a 32-bit number whose least significant 24 bits contain the starting DSP memory address and the most significant eight bits contain the data length. The data length is limited to 128 words, due to the limitation of the output buffer. If the data length is 0, 128 or greater, the PK command returns 128 words by default. When the data length equals 2, the *SimplIQ* drive regards the data as a long variable and copies it to the output buffer in the critical section, in order to ensure its integrity.

The PK command is designed to be used by the Elmo Studio and to assist qualified technical personnel in isolating software problems. PK reports in hexadecimal binary format.

Example:
PK=0x7f000200 returns the contents of the DSP memory starting at address 0x200 and continuing through word 0x280.

Attributes:	**Type:**	Status report, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Range:	[0…0x80FFFFFF]
	Unit modes:	All

The contents of a memory location may differ according to *SimplIQ* version (refer to WI[23]).

See also:
CD, WI

Reference chapter in the *SimplIQ Software Manual:*
Chapter 6, “Program Development and Execution”

PL[N] - Peak Duration and Limit

Purpose:

- PL[1] defines the motor maximum peak current, in amperes.
- PL[2] defines the motor maximum peak duration, in seconds.

This parameter is used to protect the motor (or the drive) from over-current, and to protect the load from excessive torque. The motor current (torque) command is normally limited to its peak limit, as defined by PL[1]. After a short period of torque demand higher than C[1], the torque command limit is decreased to CL[1]. If the current command has been raised to PL[1] from zero, after the time specified for peak duration (PL[2], in seconds), the motor current command will be limited to CL[1]. The motor current command remains limited to CL[1] until enough time has passed for the average requested torque command to fall below 90% of CL[1].

The LC flag indicates that the current is limited to its continuous limit.

Torque limits PL[N] and CL[N] may be changed dynamically while the motor is on.

Attributes:		
	Type:	Parameter, Real
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	PL[1]=0, PL[2]=3 (RS), Non-volatile
	Range:	PL[1]: [0…MC] PL[2]: [1…3]
	Index range:	[1, 2]
	Unit modes:	All
	Activation:	Immediate

Notes:

- Always specify a PL[1] value that can be reached. Do not specify PL[1]>V_B/R_M, where V_B is the DC motor supply voltage and R_M is the motor resistance. You should choose a PL[1] value small enough so that at peak current, there is enough voltage to drive current changes. Otherwise, at large currents, the drive speed of response will be limited by voltage saturation, and the controller performance will decrease.
- Allowed peak current may be saturated at a level lower than the PL[1] value when the PWM frequency is increased with the XP[2] command. The actual peak saturated value in amperes can be retrieved with the WS[33] command.
- The definition used here for an ampere of three-phased motor current is explained in the "Units" section of the *SimplIQ Software Manual*.
- The peak duration PL[2] specifies the time it takes to switch from the peak limit to the continuous limit, when the current PL[1] and PL[1]=MC. The actual time period for which the peak current may be applied can, however, vary significantly from PL[2].
    - If PL[1] < MC, a longer time may be allowed for the peak current, in order to protect the drive itself.

  - If, prior to the high current demand, the current demand was very close to CL[1], the switch will occur almost instantaneously.
  - If the current demand is marginally greater than CL[1], and significantly less than PL[1], the switch may take a very long time. The exact time required may be calculated from the previous formulas.
- If CL[1] ≥ PL[1], PL[1] will be the torque limit in effect at all times, and PL[2] will be ignored.
- The minimum current limit is MC/128. If PL[1] < MC/128, the PL[1] value will be accepted, but the actual current value will be limited to MC/128.
- PL[1] is used to determine the maximum and minimum limits for the PWM generator used for current reference in 50% and 100% PWM mode.

See also:
CL[N], LC, MC, TC, XP[N]

Reference chapter in the SimplIQ *Software Manual:*
Chapter 10, "The Current Controller"

PM – Profiler Mode

Purpose:
Specifies, in UM=2, if the software speed command generator applies acceleration, deceleration and smoothing limits.

The values of PM are outlined in the following table:

PM Value	Description
0	For UM=2, AC, DC and SF are not used.
1	For UM=2, AC, DC and SF are used normally.

Table 2-28: **PM Values**

PM=0 is set by the Composer program to test the step response of the speed controller. Even with PM=0, the speed command acceleration and deceleration is limited by the SD parameter.

PM=1 is the recommended value for all applications.

Attributes:	**Type:**	Command/Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=0
	Default value:	1, Non-volatile
	Range:	[0, 1]
	Unit modes:	UM=2
	Activation:	MO=1

See also:
UM, AC, DC, SD, SF

Reference chapters in the *SimplIQ Software Manual:*
Chapter 11, "Unit Modes;" Chapter 12, "The Position Reference Generator"

PP[N] - Protocol Parameters

Purpose:
Programs all communication parameters. The PP[N] command has independent fields for the parameters of all supported communication methods. These parameters are tabulated in the tables that follow.

Parameter	Description	Range
PP[1]	Type of communication. PP[1] serves as "Enter Communication Parameters" for RS-232. PP[2] and PP[4] come into effect only when PP[1] is written. The response to PP[1]=x is not the same as the response to all other commands, because the communication type switches while processing the command.	1: RS-232 2: Reserved for compatibility with previous drives.
PP[2]	RS-232 baud rate. This parameter has no immediate effect.	5: 115,200; not in use 4: 57,600 3: 38,400 2: 19,200 1: 9,600 0: 4,800
PP[3]	Spare.	
PP[4]	RS-232 parity.	0: None 1: Even 2: Odd

Table 2-29: **RS-232 Communication Parameters**

Parameter	Description	Range
PP[13]	CANopen device ID.	1 - 127
PP[14]	CAN baud rate.	0: 1,000,000 1: 500,000 2: 250,000 3: 125,000 4: 100,000 5: 50,000 6: 50,000 7: 50,000 8: 800,000
PP[15]	CAN group ID.	1 - 128

Table 2-30: **CAN Communication Parameters**

Attributes:	**Type:**	Parameter, Integer
	Source:	RS-232, CANopen
	Restrictions:	MO=0 for PP[1]
	Default values:	PP[1]=1, PP[2]=2, PP[13]=127, PP[14]=1, PP[15]=128 All others default to zero (RS), Non-volatile
	Range:	As shown in previous tables
	Index range:	[1…15]
	Unit modes:	All
	Activation:	RS-232 parameters activated by setting PP[1]. CANopen parameters activated upon restarting network through NMT service (refer to Elmo *CANopen Implementation Manual*).

Notes:

- The number of R-232 stop bits is fixed to 1.
- The group ID number for CAN (PP[15]) defines the received message object ID. The response is transmitted by each node with its own ID (PP[13]). Setting PP[15]=128 allows the user to cancel the CAN group ID.
- Unused PP[N] parameters are reserved for compatibility with other Elmo drives.

Reference chapter in the *SimplIQ Software Manual:*
Chapter 3, "Communication with the Host"

PR – Relative Position

Purpose:
Specifies that the next software position command will be a PTP (point-to-point) motion and defines its target position (refer to the PA command). PR may be applied in any active motion mode; it is activated and applies changes to the PA setting only after the next BG is executed.

- *If PTP motion is already active,* PA will be increased by the PR value and become the new position target.
- *If PTP motion is not active* and is now activated by PR, PA will be set to the new value, which is equal to the software position command plus the PR value (PA=Position command + PR). This PA value will become the new target position.

Setting the PA value always resets the PR value to zero.

Attributes:		
	Type:	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=1
	Default value:	0, Volatile. Cleared automatically at MO=1
	Range:	See Notes below
	Unit modes:	UM=3, 4, 5
	Activation:	BG

Notes:

- If a PA value was set but not activated by BG prior to activating this PR setting, the PA value will be discarded.
- The position target is calculated with modulo counting, resulting in the following range:
  [XM[1]…XM[2]] for UM=3, 5
  [YM[1]…YM[2]] for UM=4
  and restricted by VL[3] and VH[3].
  For example, in UM=5, if XM[1]=0 and XM[2]=1000, PR=2500 is similar to PR=500.
  PR is rejected if the results of PA+PR exceed the VL[3] and VH[3] limits.

See also:
PA, XM[N], YM[N]

Reference chapter in the *SimplIQ Software Manual:*
Chapter 12, "The Position Reference Generator"

PS - Program Status

Purpose:
Returns the status of the user program. If a user program is running, PS returns the number of user program threads:

- 0 if the user program is halted
- –1 if no user program thread is running
- –2 if no user program is ready to run

For detailed information about the status of the user program threads, use the DB command.

Attributes:	**Type:**	Status report, Integer
	Source:	RS-232, CANopen
	Restrictions:	None
	Unit modes:	All

See also:
CC

PT - Position Time Command

Purpose:
Specifies that the next BG will start a PT (Position - Time) tabulated motion and defines the starting index in the QP[N] array.

In a PT motion, a new position reference value is picked from the QP[N] array once per MP[4] position controller sampling times. The motion is interpolated between the points specified by the QP[N] array. The MP[N] parameters specify which indices of the QP[N] array are used for the motion. After BG, the PT motion starts from position QP[PT].

A PT motion terminates when, in non-cyclical mode (MP[3]=0), the index of MP[2] is reached, or if the PT buffer underflows (CANopen only). When a PT motion terminates, the motor is brought to a complete stop using the SD deceleration.

When a PT motion is executing, PT reads the presently executing index of the QP[N] array.

Attributes:	**Type:**	Command/Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=1, PT motion executing, Advanced model only
	Default value:	None, Volatile
	Range:	[1…1024]
	Unit modes:	Position control (UM=3, 4, 5)
	Activation:	Next BG

See also:
PV, MP[N], QP[N], QV[N], QT[N]

Reference chapter in the *SimplIQ Software Manual:*
Chapter 12, "The Position Reference Generator"

PV - Position Velocity Time Command

Purpose:
Specifies that the next BG will start a PVT (Position - Velocity - Time) tabulated motion and defines the starting index for the QP[N], QV[N] and QT[N] arrays.

In a PVT motion, new position and speed reference values are picked from the QP[N] and QV[N] arrays at the time specified by the elements of the QT[N] array. The motion is interpolated between the entries of QP[N] and QV[N]. The MP[N] parameters specify which indices of QP[N], QV[N] and QT[N] are used for the motion.

After BG the PVT motion starts from the QP[PV] position.

A PVT motion terminates when, in non-cyclical mode (MP[3]=0), the index of MP[2] is reached, or if the PVT buffer underflows (CANopen only). When a PVT motion terminates, the motor is brought to a complete stop using the SD deceleration.

When a PVT motion is executing, PV reads the presently executing index of the PVT table.

Attributes:		
	Type:	Command/Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=1, PVT motion executing, Advanced model only
	Default value:	None, Volatile
	Range:	[1…64]
	Unit modes:	Position control (UM=3, 4, 5)
	Activation:	Next BG

See also:
PT, MP[N], QP[N], QV[N], QT[N]

Reference chapter in the *SimplIQ Software Manual:*
Chapter 12, "The Position Reference Generator"

PW[N] – PWM signal parameters

Purpose:

- PW[1] defines the offset value for PWM signals in fractions of the Duty cycle.
- PW[2] defines the dead band zone of the PWM signal in the fractions of the Duty cycle.

At times, the application requires the Duty cycle value to be different from the source of the PWM signals (i.e. offset). The PW[1] parameter is intended for this purpose.

The Actual Duty Cycle is *Estimated Duty Cycle - Offset*

A very low Duty cycle values may cause a motor to "crawl" when a complete stop is desired

The dead band zone parameter (PW[2]) forces the Actual Duty Cycle to zero when it is less than PW[2].

Attributes:	**Type:**	Parameter, Real
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	PW[1]=0, PW[2]=0(RS), Non-volatile
	Range:	PW[1]: [-1.0 …1.0]
		PW[2]: [0.0 …1.0]
	Index range:	[1, 2]
	Unit modes:	All
	Activation:	Immediate

- A *SimplIQ* drive may receive the PWM signals in two formats: 50% and 100% Duty cycle. For more details, please refer to YA[] command

See also:
AN[N], FR[N], RM, YA[N]

Reference chapter in the *SimplIQ Software Manual:*

PX – Main Position

Purpose:
Reads the position of the main feedback. Upon power on, the main position is set to zero. The variable PX accumulates the main feedback pulses. PX can count cyclically (refer to the XM[N] command). When the motor is off, PX may be used to set a value for the position counter by typing PX=n. To program the position while the motor is on, refer to the HM[N] command.

Attributes:	**Type:**	Parameter/Status report, Integer
	Source:	Program, RS-232, CANopen
	Reset value:	0, Volatile
	Restrictions:	MO=0
	Range:	See Notes below
	Unit modes:	All

Notes:

- PX is limited to the position counter range.
- If PX exceeds the modulo range, it will be automatically set to a value *within* that range.
- PX counts in the direction defined by CA[16]. If CA[16] is modified, PX will not change, but position counting will be continued in the other direction. To ensure proper commutation, CA[16] must be modified in conjunction with the motor direction (CA[25]).

See also:
HM[N], XM[N], CA[N]

PY - Auxiliary Position

Purpose:
Reads the position of the auxiliary feedback. Upon power on, the auxiliary position is set to zero. The variable PY accumulates the auxiliary feedback pulses. PY can count cyclically (refer to the YM[N] command). When the motor is off, PY may be used to set a value for the auxiliary position counter by typing PY=n. To program PY while the motor is on (MO=1), refer to the HY[N] command.

Attributes:	**Type:**	Parameter/Status report, Integer
	Source:	Program, RS-232, CANopen
	Reset value:	0, Volatile
	Restrictions:	MO=0
	Range:	See Notes below
	Unit modes:	All

Notes:

- PY is limited to the position counter range.
- If PY exceeds the modulo range, it will be automatically set to a value *within* that range.
- PY counts in the direction defined by YA[5]. If YA[5] is modified, PY will not change, but position counting will be continued in the other direction. In dual feedback mode (UM=4), the motor direction must be changed accordingly.

See also:
HY[N], YM[N], YA[N], CA[N]

QP[N], QT[N], QV[N] - Position, Time, Velocity

Purpose:
Stores data for the PT and PVT motion modes. The QP[N], QV[N] and QT[N] arrays define the position (QP[N]) and speed (QV[N]) at any single time instance (QT[N]).

With CANopen communication, the QP[N], QV[N] and QT[N] arrays can be programmed at high speed using specially designed communication objects.

Attributes:	**Type:**	Command/Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default value:	0, Volatile
	Range:	QP[N]: [-2^{31}…2^{31} – 1]
		QT[N]: [1…255], in msec
		QV[N]: [-2^{31}…2^{31} – 1]
	Index range:	QP[N]: [1…1024]
		QT[N]: [1…64]
		QV[N]: [1…64]
	Unit modes:	[3…5]
	Activation:	Immediate

The values of QP[N] and QV[N] are unlimited, to enable their use as general-purpose storage when PVT/PT is not used. However, for PVT/PT, the QP[N] and QV[N] values should be confined to the position and velocity ranges.

See also:
PT, PV, MP[N]

Reference chapter in the *SimplIQ Software Manual:*
Chapter 12, "The Position Reference Generator"

RC - Define Recorded Variables

Purpose:
Defines which signals are to be recorded.

The drive can record a range of signals for performance verification and debugging. The first step of the recording process is the definition of the recorded variable by assigning a value to RC, a bit field. Each "on" bit in the binary representation of RC defines a signal to be recorded. The host can map many optional variables to any bit of RC from bit 0 to bit 15.

A valid RC defines at least one recorded variable. Up to eight variables can be selected.

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	Recorder inactive (RR=0 or RR=-1)
	Default value:	0 (RS), Non-volatile
	Range:	See previous description
	Unit modes:	All
	Activation:	Next initiation of recorder (RR)

Notes:

- If the drive has stored a previously-recorded data, setting RC will invalidate this data. Invalidated data cannot be retrieved.
- The total number of data points that may be recorded is fixed. Therefore, the number of points per signal depends on the number of signals recorded simultaneously: the more signals recorded, the fewer points available for each signal.

See also:
RG, RL, RP[N], RR, BH

Reference chapter in the *SimplIQ Software Manual:*
Chapter 7.4, "The Recorder"

RG - Recorder Gap

Purpose:
Defines the frequency per sampling times that the recorder is activated.

Because the recorder has a limited storage capacity, if it operates at the sampling time of the drive, the recorder will operate for a very short time. For longer recording times, the time interval between consecutive data recordings must be increased. The RG parameter trades recording resolution against recording time. With RG=1, the sampling time of the recorder is WS[29]. Be aware that recorder sampling time depends on the TS value and the specific unit mode (UM).

Attributes:		
	Type:	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	Recorder inactive (RR=0 or RR=-1)
	Default value:	1 (RS), Non-volatile
	Range:	[1…4096]
	Unit modes:	All
	Activation:	Next initiation of the recorder (RR)

If the drive has stored a previously recorded data vector, setting RG will invalidate this data. Invalidated data cannot be retrieved.

See also:
RC, RL, RP[N], RR, BH, TS, WS[29], UM

Reference chapter in the *SimplIQ Software Manual:*
Chapter 7.4, "The Recorder"

RL - Record Length

Purpose:
Specifies the length of the recorded data, as follows:

Number of Simultaneously Recorded Signals	Maximum Record Length
1	4096
2	2048
3	1365
4	1024

RL can specify that the signal records will be shorter than the maximum. If RL is set to a value higher than the maximum record length (for example, RC defines three recorded signals, but RL=2048), the recorder will still work. The length of the records, however, will be shorter than RL. The actual size of the recorded data is returned by WI[21].

Attributes:

Type:	Parameter, Integer
Source:	Program, RS-232, CANopen
Restrictions:	Recorder inactive (RR=0 or RR=-1)
Default value:	256 (RS), Volatile
Range:	[1…4096]
Unit modes:	All
Activation:	Next initiation of the recorder (RR).

If the drive has stored a previously recorded data vector, setting RL will invalidate this data. Invalidated data cannot be retrieved.

See also:
RC, RG, RP[N], RR, BH

Reference chapter in the *SimplIQ Software Manual:*
Chapter 7.4, "The Recorder"

RM - Reference Mode

Purpose:
Specifies the use of an external reference signal. In all unit modes, the *SimplIQ* drive sums the reference command to the drive from two sources:

- A software command, generated internally either by a communicated command or by a user program
- An "auxiliary" reference, may be the sum of analog input #1 and the external signals that are derived by the auxiliary encoder (possibly using the ECAM table) or PWM signals.
- In position mode (UM=5), the analog input is not used as a reference.
- In stepper mode (UM=3), an auxiliary reference is sets the torque value to the stepper generator.

The RM values are as follows:

RM Value	Meaning
0	The reference command is generated internally, either by the interpreter command or by the user program.
1	The reference command is summed from the internal software reference command and the auxiliary reference command.

Attributes:

Type:	Parameter, Integer
Source:	Program, RS-232, CANopen
Restrictions:	Defined in following notes
Default value:	0 (RS), Non-volatile
Range:	[0, 1]
Unit modes:	All
Activation:	MO=1

Notes:

- If you are not using the auxiliary referencing option, set RM=0 to prevent A/D noise from falsely driving the drive.
- In dual loop mode (UM=4), switching to auxiliary reference mode (RM=1) is not allowed.
- In position mode (UM=5), RM may be changed only when the software reference is stationary (MS ≤ 1). In speed mode (UM=2) RM may be changed at any time.

See also:
UM, FR[N], AG[N], AS[N], PW[N]

Reference chapter in the *SimplIQ Software Manual:*
Chapter 11, "Unit Modes"

RP[N] - Recorder Parameters

Purpose:
Enables the complete specification of how the recorder is triggered and how the recorded data is transferred to the host.

Trigger definitions:
The recorder is started by a trigger event, which may be one of the following:

- *Immediate:*
  The recorder starts immediately after the recording request has been issued.
- *Triggered by an analog signal:*
  The recorder starts upon one of the following events:
  - Positive slope: The signal crosses a prescribed level with a positive slope
  - Negative slope: The signal crosses a prescribed level with a negative slope
  - Window: The signal exits a window of two prescribed signal levels.
  - Digital inputs: Digital inputs are switched to their active logic state as defined by the IL[N] command.
- *Motion begins:*
  A BG command, or the activation of a hardware BG command (refer to the IL[N] command).

Trigger delay:
The trigger defines when the recorder is to start. The recorder can be programmed to start before the trigger event, so that the trigger event can be caught "in the middle of the action." This is possible because the recorder starts to record at the instant it is launched by the RR command, so that when the trigger event occurs, the pre-trigger information is already recorded. The trigger parameters are listed in the following table:

RP[N]	Range	Definition
RP[0]	[0…1]	0: Time quantum is 4*TS 1: Time quantum is TS
RP[1]: Trigger variable	[1…65,536]	Defined similarly to RC, but only 1 bit may be non-zero. The trigger variable does not need to be one of the recorded variables.
RP[2]: Pre-trigger storage in percent	[0…100]	The percentage of the recorded signal taken before the trigger event.
RP[3]: Trigger type	[0…5]	0: Immediate 1: BG 2: Positive slope 3: Negative slope 4: Window 5: Digital inputs

RP[N]	Range	Definition
RP[4]: Level 1	Unlimited	Level for positive slope trigger, or high side for window trigger.
RP[5]: Level 2	Unlimited	Level for negative slope trigger, or low side for window trigger.
RP[6]: Polarity	[0…63]	Defines the polarity for the digital input trigger of the recorder. 1: Positive polarity.
RP[7]: Digital input mask	[0…63]	Defines which digital inputs trigger the recorder.

Table 2-31: **Trigger-related RP[N] Parameters**

The following parameters enable the BH command to fetch only the required part of the recorder results:

RP[N]	Range	Definition
RP[8]: Index low	[0…1024]	Lower buffer index for recorded data transmission.
RP[9]: Index high	[0…1024]	Higher buffer index for recorded data transmission. When RP[9]=RP[8]=0, all of the buffer is transmitted.

Attributes:

Type:	Parameter, Integer
Source:	Program, RS-232, CANopen
Restrictions:	None
Default values:	RP[1] is set by the UM command. The other RP[N] values are defaulted to 0 at power on
Range:	Defined in previous tables
Index range:	[1…9]
Unit modes:	All
Activation:	Trigger parameters: RR=3 Upload parameters: Immediate

If the drive has stored a previously recorded data vector, setting RP[N] (with N other than 8 or 9) will invalidate this data. Invalidated data cannot be retrieved.

See also:
RC, RG, RL, RR, BH

Reference chapter in the *SimplIQ Software Manual:*
Chapter 7.4, "The Recorder"

RR - Activate Recorder / Get Recorder Status

Purpose:
Launches the recorder, kills an on-going recording process or retrieves the recorder status. The RR command has the following options:

RR Value	Meaning
0	Kill the recorder (do nothing if the recorder is not active).
1	Start recording at the next BG command.
2	Start recording immediately.
3	Arm the recorder with the trigger settings of the RP[N] parameters.

Table 2-32: **RR Command Options**

As a status report, RR may return the following values:

RR Report	Meaning
-1	There is no valid data in the recorder.
0	The recorder action is complete and it is loaded with valid data.
1 - 3	Waiting for the completion of RR=1, RR=2 or RR=3, respectively. The report value 3 does not indicate if the recorder is already recording or just waiting for a trigger. If this differentiation is required, use the SR command.

Table 2-33: **RR Reports**

Attributes:

Type:	Parameter, Integer
Source:	Program, RS-232, CANopen
Restrictions:	None
Default value:	-1, Non-volatile
Range:	[0…3]
Unit modes:	All
Activation:	Immediate

See also:
BH, RP[N], RC, RG, RL, RR

Reference chapter in the *SimplIQ Software Manual:*
Chapter 7.4, "The Recorder"

RS - Soft Reset

Purpose:
Initializes the drive parameters to their factory default, and resets all volatile variables to their power-on default.

Attributes:	**Type:**	Command, No value
	Source:	RS-232, CANopen
	Restrictions:	MO=0, Program not running
	Unit modes:	All
	Activation:	Immediate

Notes:

- RS does not change the communication settings; therefore, after executing RS, it is still possible to communicate with the drive. The communication parameters, however, are reset. For example, if the baud rate is 9600, setting PP[1]=1 immediately after RS will switch the baud rate to the default of 19,200.
- The RS command disables the communication routines for a few milliseconds. If an RS is executed by an RS-232 command, a CAN message may be lost in the execution interval.
- The RS command modifies only the RAM contents; it does not affect the flash memory. Use the SV command to make the effect of RS permanent.
- After an RS command, the current limits are set to zero so it is impossible to start the motor immediately. After an RS, it is recommended to go through the steps of the Composer Wizard before attempting to work with the motor.

See also:
LD, SV

RV[N] - Recorded Variables

Purpose:
Maps recorded variables to the recorder through the RC command. By setting RV[N]=X, bit N-1 of RC is assigned the X variable in the variable static table. The default mapping (power on) of RV[1] - RV[16] behaves similarly to previous projects. The full list of variables available to the recorder is stored in the serial flash memory of the *SimplIQ* drive and can be uploaded using the LS command.

Attributes:	**Type:**	Parameter, Integer, Bit-field
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	RV[1]=1, RV[2]=2 . . . RV[16]=16 (RS), Volatile
	Range:	According to static table variable index.
	Index range:	[1…16]
	Unit modes:	All
	Activation:	Immediate

See also:
RC, BH, RR

Reference chapters in the *SimplIQ Software Manual:*
Chapter 8, "The Recorder;" Appendix A, "The *SimplIQ* Flash Memory Organization"

SD – Stop Deceleration

Purpose:
Defines the deceleration in counts/second^2 used to stop motions in case of emergency. In addition, SD defines the acceleration limit for the combination of software and external reference commands.

Position-controlled motions cannot be stopped abruptly, because:

- The discontinuity in the reference speed may produce position errors in excess of ER[3]. This cuts the motor drive to freewheeling, which may be a safety problem with high inertia or fast-moving loads.
- If the maximum allowed acceleration value (GS[9]) is not correctly set, generating a large position error can destabilize the position controller.

The SD parameter must be defined as the largest deceleration in which the motor can accelerate/decelerate to the load.

The SD acceleration is applied in the following cases:

- An ST command or hardware stop command (refer to the IL[N] command)
- Data underflow in a PT or PVT motion
- Detection of a limit switch or an abort switch

Attributes:		
	Type:	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=0
	Default value:	1,000,000,000 (RS), Non-volatile
	Range:	Stop deceleration range
	Unit modes:	UM=2, 3, 4, 5
	Activation:	MO=1

See also:
AC, DC, ST

Reference chapter in the *SimplIQ Software Manual:*
Chapter 11, "Unit Modes"

SF - Smooth Factor

Purpose:
Defines the motion smoothing factor for PTP and jogging motions. Smoothing means that the motion speed profile has no "sharp corners." The price for smoothing is that the total time required for completing the motion increases. For SF>0, the acceleration to the required speed is not set immediately to its final value but takes SF milliseconds to build. The total time required to complete the motion is increased by SF milliseconds. The maximum SF for the actual sampling time, in milliseconds, is given by WI[22].

Attributes:		
	Type:	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=0
	Default values:	0 (RS), Non-volatile
	Range:	[0…100]; See Notes below
	Unit modes:	UM=2, 3, 4, 5
	Activation:	BG

Notes:

- WI[22] reports the maximum permitted value for SF. Setting SF greater than WI[22] is not reported as an error, but the smoothing factor that is applied may be less than the actual setting. WI[22] depends on the sampling time (TS command).
- The smoothing action applies for PTP and jogging motions only. SF does not affect tabulated motions such as PT and PVT.
- If a PT/PVT motion command is given while a smoothed motion is on, a jump in the software reference may occur due to the removal of the smoothing filter. The stop manager will limit the rate in which the position command catches up with the software command that has jumped.
- If a PTP or jog motion command is given while PT/PVT motion is on, the smoothing effect will gradually build up, until complete smoothing is reached after SF milliseconds.
- SF does not affect the external velocity or position reference.

See also:
AC, DC, JV, SP

Reference chapters in the *SimplIQ Software Manual:*
Chapter 11, "Unit Modes;" Chapter 12, "The Position Reference Generator"

SN – Serial Number

Purpose:
Returns the contents of CANopen object 0x1018 (LSS protocol) as an integer. The LSS protocol defines the behavior of the CAN node for ID setting and baud rate changing. Object 0x1018 includes the identification of the specific CAN node in such a way that the identification is totally unique.

The identification number contains four entries represented in SN[N] as follows:

- SN[1] returns the vendor ID.
- SN[2] returns the product code.
- SN[3] returns the revision number.
- SN[4] returns the serial number.

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default value:	0, Volatile
	Range:	32 bits
	Index range:	[1…4]
	Unit modes:	All
	Activation:	Immediate

SP - Speed for PTP Mode

Purpose:
Sets the maximum speed for PTP (point-to-point) motion. At the start of motion, the speed of SP is reached with the acceleration of AC. Then, a constant speed of SP is maintained until the deceleration to final stop begins with the DC acceleration.

The speed of SP counts/second is achieved only if the motion is long enough, if AC and DC are large enough, and if SF is small enough. Otherwise, the speed limit of SP remains inactive.

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	|VL[2]| ≥ SP and VH[2] ≥ SP
	Default value:	25,000 (RS), Non-volatile
	Range:	Velocity range
	Unit modes:	UM=3, 4, 5
	Activation:	BG

See also:
HL[N], LL[N], AC, DC, SF, PA, PR

Reference chapter in the *SimplIQ Software Manual:*
Chapter 12, "The Position Reference Generator"

SR - Status Register

Purpose:
Returns a bit-field, reporting the status of the system in a concise format. Most of the information in SR may be recovered using other commands. The primary purpose of the SR command is to enable a remote host — such as the Composer program — to get a snapshot of the system status without overloading the communications system.

Reported Status	Bits
Drive read: 0: Conditions OK 1: Problem, as reported by bits 1 - 3	0
Servo drive status indication details: refer to the following table	1 - 3
Motor on (MO)	4
Reference mode (RM)	5
Motor failure latched (see MF for details)	6
Unit mode (UM)	7 - 9
Gain scheduling on	10
Either Main or Auxiliary Homing being processed	11
Program running	12
Current limit on (LC)	13
Motion status reflection (MS)	14 - 15
Recorder status: 0: Recorder inactive, no valid recorded data 1: Recorder waiting for a trigger event 2: Recorder finished; valid data ready for use 3: Recording now	16 - 17
Not used	18 - 23
Digital Hall sensors A, B and C[5]	24 - 26
CPU status: 0: CPU OK 1: Stack overflow or CPU exception	27
Stopped by a limit – RLS, FLS, Stop switch – or by a VH[3]/VL[3] position command limit	28
Error in user program	29
Unused	30 - 31

Table 2-34: **Status Register Bits**

[5] The digital Hall sensor readings in SR are corrected according to CA[1…6].

0x8	0x4	0x4	Meaning
0	0	0	OK.
0	0	1	Under voltage: The power supply is shut off or it has too high an impedance.
0	1	0	Over voltage: The power supply voltage is too large, or the servo drive did not succeed in absorbing the kinetic energy while braking a load. A shunt resistor may be needed.
1	0	1	Short circuit: The motor or its wiring may be defective.
1	1	0	Temperature: The drive is overheating.

Table 2-35: **Servo Drive Status Indications**

Attributes:	**Type:**	Status report, Integer, Bit-field
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Unit modes:	All

See also:
MF, RR, PM, UM, RM, IP, MS, MO, LC, HM, HY

ST - Stop Motion

Purpose:
Stops the software motion. The software commands decelerate to a complete stop using the SD deceleration. ST does not affect the external position reference and has no effect for the following conditions: MO=0 and UM=1.

Attributes:	**Type:**	Command, No value
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Unit modes:	UM=2, 3, 4, 5
	Activation:	Immediate

See also:
BG, RM

Reference chapter in the *SimplIQ Software Manual:*
Chapter 12, "The Position Reference Generator"

SV - Save Parameters to Flash

Purpose:
Saves the entire set of non-volatile variables from the RAM to the flash memory. Before saving, the parameter integrity is tested. If the test fails, the SV command exits with an error and the flash contents remain as is. The CD command details the reason for the failure.

Attributes:	**Type:**	Command, No value
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=0, User program not running
	Unit modes:	All
	Activation:	Immediate

Notes:

- The SV command may take a few hundreds of milliseconds to execute, during which the communication drivers are disabled. If an SV command is executed by an RS-232 command, a CAN message may be lost in the interim, and vice versa.
- The SV command does not store the commutation tables; refer to the HV[N] command.
- Be aware that the flash memory is limited by the number of times data may be saved to it (about 100,000 saves). Therefore, be very careful when defining the use of SV in a user program.

See also:
CD, HV[N], LD

SY - Say Something

Purpose:
Enables the drive to respond with a printed comment.

Attributes:	**Type:**	Command, No value
	Source:	RS-232, CANopen
	Restrictions:	None
	Unit modes:	All
	Activation:	Immediate

TC - Torque Command

Purpose:
Sets the torque (motor current) command, in amperes, for the torque-control software-reference modes (UM=1 and UM=3). TC commands are accepted in the range permitted by the present torque command limits (refer to the PL[N] and CL[N] commands).

If TC is set greater than CL[1], after a few seconds, the current limit of the servo drive will drop to CL[1].

TC defines the reference value IQ (the ID command is always zero). When RM=1, TC is summed with the external analog commands.

Attributes:	**Type:**	Command/Parameter, Real
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=1, UM=1 or UM=3
	Default value:	0, Volatile. Cleared automatically at MO=1
	Range:	Torque limits
	Unit modes:	UM=1, 3
	Activation:	Immediate

See also:
MO, UM, IQ, ID, CL[N], PL[N], MC

Reference chapters in the *SimplIQ Software Manual:*
Chapter 11, “Unit Modes;” Chapter 14, ”Limits, Protections, Faults and Diagnosis”

TM - System Time

Purpose:
Reads and writes the system time, in microseconds. *SimplIQ* drives have a 32-bit microsecond counter. In the absence of CAN SYNC and TSTAMP signals, the microsecond counter runs freely, completing a cycle approximately once per 71.5 minutes. The CAN SYNC and TSTAMP sequence synchronize this counter to the microsecond counter of the network master (refer to the *SimplIQ CAN Implementation Manual*).

The BT command uses the TM variable to coordinate the start of motion. The tdif( ) function uses TM to calculate time differences.

Notes:

- When CAN communication is used, setting or updating the system counter by the user may upset the synchronization with the CAN network master.
- The tick system function also reads the system time of the *SimplIQ*, but its value differs from the TM command value.

Attributes:	**Type:**	Command/Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default value:	None, Volatile
	Unit modes:	All
	Activation:	Immediate

Examples:

QP[1023]=TM	Read TM into a software variable (here, QP[1023] is used as general storage, not as motion reference).
tdif(QP[1023])	Return the time elapsed since TM was sampled into QP[1023].
TM=0	Reset system time to zero.

The time returned by this command may jump due to a CAN master update (refer to the Elmo *CANopen Implementation Guide.*

See also:
BT

Reference chapter in the *SimplIQ Software Manual:*
Chapter 4, ”The Interpreter Language” (tdif function)

TP[N] - Floating Wizard Parameters

Purpose:
Contains parameters for internal use only.

Attributes:	**Type:**	Report, Floating
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Range:	Not applicable
	Unit modes:	All
	Activation:	Immediate

TR - Target Radius

Purpose:
Provides the criterion for deciding that a motion is complete and the motor is stabilized in place with the required accuracy, defined in terms of target radius and target time. The target radius is the maximum positioning error allowed for static stabilization (not to be confused with the ER[N] parameters, which represent the dynamic stabilization error that is considered a fault). The target time is the minimum time the absolute value of the error must be within the target radius in order to determine that the motor is stabilized in place after PTP (point-to-point) motion.

- TR[1] defines the target radius in counts.
- TR[2] defines the target time in milliseconds.

When a target position range is within the target radius (TR[1] command) for at least the target time (TR[2] command), the MS status is set to 0.

The TR[1] value must be within limits of the modulo of the position feedback: (XM[1]…XM[2]) for UM=5 and (YM[1]…YM[2]) for UM=4.

Attributes:		
	Type:	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	TR[1]=100, TR[2]=20 (RS), Non-volatile
	Range:	TR[1]: [0…32,000], TR[2]: [0…100]
	Unit modes:	UM=4, 5
	Activation:	Immediate

Reference chapter in the *SimplIQ Software Manual:*
Chapter 12, "The Position Reference Generator"

TS - Sampling Time

Purpose:
To define the sampling time of the drive, in microseconds. TS is the sampling time of the current loop. The sampling time of the velocity controller is two times TS and the sampling time of the position controllers (UM=4, 5) is four times TS. For example, if TS=80, the torque/commutation controller runs once every 80 microseconds, the speed controller executes every 160 microseconds, and the position controller executes every 320 microseconds.

The selection of TS is a compromise between high servo performance and the scan loop (background) operations, such as user program and interpreter responses. A low TS enables the drive to achieve more control bandwidth, but at the same time, it increases the computational burden on the CPU, so that less computing power remains for executing interpreter and user program commands.

The drive does not allow an excessively low value for TS to prevent an overflow of the required CPU computing power.

For all unit modes, WS[28] gives the actual sampling time of the speed controller and WS[55] gives the actual sampling time of the position controller.

Notes:

- When TS is modified, the current loop gains must be retuned in order to prevent instability of the current loop, which may damage the drive and/or the motor.
- Too short a TS value will cause a CPU stack overflow. If a stack overflow occurs, change TS in order to start the motor and reset the CPU stack overflow errors. Setting a new value for TS removes all stack overflow history. This should be avoided.
- Speed limit values (VH[2],VL[2],HL[2],LL[2]) depend on TS value (refer to the LL command for mor info).
- Use WI[7] to measure the CPU load. WI[7] returns the CPU percentage remaining for interpreting user commands and running the user program. Use WI[7] in true working conditions, because the CPU load changes with the order of the control filter and other working conditions, like the motion mode does.
- The smooth factor range (SF command) depends on the range of TS. The actual SF can be retrieved by WI[22].

Attributes:		
	Type:	Parameter, Integer
	Source:	RS-232, CANopen
	Restrictions:	MO=0, User program not running
	Default value:	90 (RS), Non-volatile
	Range:	[70…120]
	Unit modes:	All
	Activation:	Immediate

See also:
WI[N], WS[N]

TW[N] - Wizard Command

Purpose:
Contains parameters for internal use only. For example, the command is used for auto-tuning or debugging.

Attributes:	**Type:**	Report, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Range:	Not applicable
	Index range:	[1…32]
	Unit modes:	All

UF[N] – User Float array

Purpose:
Provides an array of 24 floating numbers for general-purpose use.

Attributes:	**Type:**	Parameter, Float
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	0 (RS), Non-volatile
	Range:	[-1e20…1e20]
	Index range:	[1…24]
	Unit modes:	All
	Activation:	Immediate

Typical applications:
General look-up tables of real numbers and parameters for machine task definition.

Example:
See example of UI[]

See also:
UI

UI[N] – User Integer

Purpose:
Provides an array of 24 integer numbers for general-purpose use.

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default values:	0 (RS), Non-volatile
	Range:	$[(-2^{30}+1)\ldots(2^{30}-1)]$
	Index range:	[1…24]
	Unit modes:	All
	Activation:	Immediate

Typical applications:
General look-up tables of real numbers and parameters for machine task definition.

Example:
```
MO=0;
int MY_VAR ;
MY_VAR=10 ;
UI[2]=MY_VAR ;
for UI[1]=1:10
UF[UI[1]]=UI[2]*SP/UI[4];

end
```
In this example, the UF[] array is dynamically indexed and valued by elements of the UI array.

See also:
UF

UM - Unit Mode

Purpose:
Defines the motion controller drive configuration, as follows:

UM Value	Description (Related Commands)
1	Torque control mode In this mode, the motor current command is set directly by the TC software command or by an analog reference signal. This mode is useful for torque or force control when the servo drive is used only as an inner device within an external feedback loop.
2	Speed control mode In this mode, the motor speed command is set directly by the JV software jogging command or by an analog reference signal.
3	Micro-stepper mode In this mode, no commutation is made. The user controls the electrical field angle using the position commands and the motor current (holding torque) by the TC command or by an analog signal. All position mode commands (PA, JV or tabulate motion) are applicable. This mode cannot be used with DC motors.
4	Dual feedback position control In this mode, the position controller stabilizes the position of the auxiliary feedback input. The position controller issues a motor speed command to an inner speed control loop. The inner speed control loop derives its speed feedback from the main feedback input. In this case, auxiliary reference mode (RM=1) is not allowed. The position command is generated similarly to UM=5, except that the auxiliary feedback is not available for reference generation.
5	Single loop position control In this mode, the position controller stabilizes the position of the main feedback input. The position command is summed by a software command — point-point (PA), jogging (JV) or tabulated motion (PT, PVT) — and from an external command. The external command is derived from the auxiliary feedback reference (FR[N], ET[N], EM[N]).

Table 2-36: **UM Values**

The unit mode is reflected in the SR report.

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=0, For UM=4, Advanced model only For UM=4, RM=1 For UM=2, 4/5, for Banjo and Trumpet
	Default value:	3 (RS), Non-volatile
	Range:	[1…5]
	Activation:	Immediate

See also:
RM, SR, TS, WI[N]

Reference chapter in the *SimpliQ Software Manual:*
Chapter 11, "Unit Modes"

VE - Velocity Error

Purpose:
Reports the present velocity tracking error:
VE = DV[2] – VX

If the absolute value of VE exceeds ER[2], motion is aborted and motion fault code MF=128 (0x80) is set.

If MO=0, or if the speed controller is not used (UM=1, 3), VE returns 0.

Attributes:	**Type:**	Status report, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Unit modes:	UM=2, 4, 5

VH[N], VL[N] - High and Low Reference Limit

Purpose:
Define the drive's minimum and maximum speed and position reference limits.
Software commands beyond these values are not accepted, and are truncated to VL[N] and VH[N].

Speed limits are restricted by the TS value according to the following:

For analog encoders, resolvers, tachometers, potentiometers and digital halls:

SpeedLimit = minimum between 80,000,000 and 8e9/TS.

For quadrature encoder:

SpeedLimit = minimum between 20,000,000 and 8e9/TS.

- The reference to the speed controller is limited to the [VL[2]…VH[2]] range.
- The reference to the position controller is limited to the [VL[3]…VH[3]] range.

Attributes:	**Type:**	Command/Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=0 VH[N] > VL[N] for all entries
	Default values:	VH[2]=15,000,000, VH[3]= 999,999,990 VL[2]=-15,000,000, VL[3]=-999,999,990 (RS), Non-volatile
	Range:	VH[2]: Velocity range VL[2]: Velocity range $-2^{31} \leq VH[3] \leq 2^{31} - 1$ $-2^{31} \leq VL[3] \leq 2^{31} - 1$
	Unit modes:	VH[2], VL[2]: UM=2, 3, 4, 5 VH[3], VL[3]: UM=3, 4, 5
	Activation:	Immediate

In position modes (UM=4, 5) motor movement is enabled in both directions *within* the defined position reference range. If feedback has been extended beyond those limits, the motor can be enabled by the user (MO=1) but the motion can only be in the direction *towards* the reference limit range.

VH[2] is used to determine the maximum and minimum allowed speed for PWM reference generator in 50% and 100% PWM modes.

See also:
XM[N], LL[N], HL[N], RM

Reference chapter in the *SimplIQ Software Manual:*
Chapter 14, "Limits, Protections, Faults and Diagnosis"

VR – Firmware Version

Purpose:
Reports the version of the firmware as a string, which includes:

- The product name
- The software version
- The software release date

This command is intended for use only by RS-232 communication, or CAN OS.

Attributes:	**Type:**	Status report, String
	Source:	RS-232
	Restrictions:	None
	Unit modes:	All

Examples:
VR
Harmonica 1.01.02.03 1Jan2002
where:

- The product name is Harmonica.
- The software version is 1.01.02.03.
- The software release date is 1 January 2002.

VX, VY - Velocity of Main and Auxiliary Feedback

Purpose:

- The VX status report returns the speed of the main feedback, in counts/second.
- The VY status report returns the speed of the auxiliary feedback, in counts/second.

The VX and VY signals are calculated using the time difference measured between consecutive feedback pulses.

Attributes:	**Type:**	Status report, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Unit modes:	All

WI[N] - Miscellaneous Reports, Integer

Purpose:
Reports integer constants and variables of the system usually used by the Composer program rather than directly by the users.

Index	Description
WI[1]	Minimum sampling time for torque control mode.
WI[2]	Reserved.
WI[3]	Reserved.
WI[4]	Maximum sampling time for torque control mode.
WI[5]	Reserved.
WI[6]	Reserved.
WI[7]	Available CPU time for background tasks, remaining after real-time control algorithm performance.
WI[8]	First amplitude for the auto-phasing process.
WI[9]	Second amplitude for the auto-phasing process.
WI[10]	First oscillation phase for the auto-phasing process.
WI[11]	Second oscillation phase for the auto-phasing process.
WI[12]	Oscillation phase average for the auto-phasing process.
WI[13]	Integer that shows entire estimated sampling time, in microseconds.
WI[14]	Integer (unsigned short) that shows a fraction of estimated sampling time, in microseconds. The value of the estimated sampling time is calculated according to the following formula: WI[13] + WI[14]/2^16.
WI[21]	Actual length of each recorded vector.
WI[22]	Maximum valid value of SF for current TS value.
WI[23]	Returns 1 if the Harmonica is running from the RAM and 0 if it is running from the flash.

Table 2-37: **WI[N] Report Details**

Attributes:

Type:	Command/Status report, Integer
Source:	Program, RS-232, CANopen
Restrictions:	None
Unit modes:	All

See also:
WS[N]

Reference chapters in the *SimplIQ Software Manual:*
Chapter 8, "The Recorder;" Chapter 9, "Commutation"

WS[N] - Miscellaneous Reports

Purpose:
Provides certain conversion constants and internal states of the drive. WS[N] gives service personnel a fairly comprehensive report, but these details are not normally required for defining an application.

The following summarizes WS[N] reports. Indices omitted from the table are not used.

Index	Report
WS[3]	CPU clock frequency, in Hz.
WS[4]	Width of the PWM frame, in CPU clocks.
WS[5]	A/D bits per one ampere of phase motor current. The A/D resolution is 1/WS[5] ampere.
WS[8]	State of the peak/continuous filter.
WS[10]	Applicable torque limit, in torque command units (see WS[22]).
WS[11]	Returns 1 for drives that support position control, and 0 for drives that support speed control only.
WS[12]	Shoot-through delay, in CPU clocks.
WS[15]	Motor oscillation actual amplitude, for an auto-phasing process.
WS[16]	Reports 1 if previous auto-phasing process failed due to an excessive transfer function phase.
WS[17]	Reports RMS of reference waveform, for automatic current controller tuning.
WS[18]	Reports mean power supply voltage, for automatic current controller tuning.
WS[19]	Reports current following error, for automatic current controller tuning.
WS[20]	Reports the stator field angle, 1024 counts per electrical revolution.
WS[21]	Reports the commutation counter.
WS[22]	Reports the scale between torque commands, in amperes, and their internal representation.
WS[28]	The sampling time of the speed controller.
WS[29]	The basic time quanta for the recorder, in microseconds.
WS[30]	Product capabilities and hardware configuration string. See the following table. The WS[30] report helps the setup program to identify the product. WS[30] has the same format for all *SimplIQ* products.
WS[33]	Reports the actual peak current saturation in amperes.
WS[55]	The sampling time of the position controller.
Other indices	Report specifically to the automatic controller tuning process.

Table 2-38: **WS[N] Miscellaneous Reports**

WS[30] is a bit-field.
The bit descriptions of WS[30] are summarized in the following table:

Bits	Meaning	
0…4	**Value**	**Product**
	0	Saxophone
	1	Clarinet
	2	Reserved
	3	Harmonica
	4	Cello
	5	Bassoon
	6	Trumpet
	7	Tuba
	8	Banjo
	9	Cornet
	10-31	Reserved
5…7	**Value**	**Main Position Sensor**
	0	Reserved
	1	Incremental digital encoder
	2	Resolver
	3	Incremental digital or analog encoder
	4	Tachometer or Potentiometer
	5	Reserved
	6	Reserved
	7	Special treated main feedback look at bits 21..24
8…10	**Value**	**Auxiliary Position Sensor**
	0	Reserved
	1	Incremental digital encoder
	2	Resolver
	3	Incremental digital or analog encoder
	4	Special treated auxiliary feedback look at bits 25..28
	5-7	Reserved
11…15	**Value**	**Grade**
	0	Standard
	1	Advanced
	12-15	Reserved
16	**Value**	**Current Controller**
	0	Analog
	1	Digital
17	**Value**	**CAN Communication**
	0	Not present
	1	Present
18 - 21	**Value**	**Special Main Feedback Configuration**
	0	None
	1	Absolute Encoder + Incremental digital/analog interface
	2-15	Reserved
22 - 25	**Value**	**Special Aux Feedback Configuration**
	0	None
	1..:15	Reserved
26 – 31	Reserved	

Table 2-39: **WS[30] Drive Personality Descriptor**

Attributes:	**Type:**	Report, Floating
	Source:	RS-232, CANopen
	Restriction:	None
	Range:	Not applicable
	Unit modes:	All

Reference chapters in the *SimplIQ Software Manual:*
Chapter 8, "The Recorder;" Chapter 9, "Commutation;" Chapter 10, "The Current Controller"

XA[N] - Extra Parameters (more)

Purpose:
Extra filters parameters. This command is used internally to allow more filtering capabilities in the current loop.

Index	Value Range	Meaning
0		Not used
1	1…32767.	Proportional value to the relative closed loop overshoot due to step input command.
2	1…32767	Proportional value to the relative closed loop overshoot due to ramp input command.
3	0…100	Approximately the level of relative overshoot over MC.
4		Reserved

Table 2-40: **XA[N] Entries and Values**

Notes:

- Do not modify the XA[] values. These values are automatically programmed into the drive during current loop tuning.

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=0
	Default values:	0
	Range:	Previous table
	Unit modes:	All
	Activation:	Immediate

Reference chapter in the *SimplIQ Software Manual:*

XC, XQ - Execute or Continue Program

Purpose:
Executes the user program from a specified label, or runs a specified function.

- XQ##MYFUNCTION(a,b,c) runs the function MYFUNCTION(a,b,c).
- XQ##LABEL runs from ##LABEL.
- XQ## runs from the start of the user program code.
- The XQ command without a parameter is illegal.
- XQ does not return a value.

The XQ command clears the error status of the program, along with run-time error flags. It does not reset program variables and does not clear the interrupt mask. XC continues a halted program (refer to the HP command).

Attributes:	**Type:**	Command, String
	Source:	RS-232, CANopen
	Restrictions:	Program loaded, compiled and not running. XC requires a running program to be halted.
	Unit modes:	All
	Activation:	Immediate

Reference chapters in the *SimplIQ Software Manual:*
Chapter 6, "Program Development and Execution"

XM[N] - X Modulo

Purpose:
Specifies the counting range for the main feedback, which is [XM[1]…XM[2]-1].

The position of the main feedback is always counted cyclically. This means that after the position is counted to its maximum value, the next position count will reset the position counter back to its minimum value. The speed reading is not affected by the position jump. **For example:** If XM[1]=-5 and XM[2]=5, the main position is counted in a cycle length of 10. The main position will always be in the range [-5…4]. If the main feedback rotates in the positive direction, the main position count will proceed from 0, 1, 2, 3, 4 to -5, -4, -3, -2, -1, 0, 1 . . . and so on.

If XM[N] is nonzero with UM=5, the controlled motor position will count cyclically. All motion trajectories will go the short way. **For example:** If XM[1]=-512, XM[2]=512, the initial position is PX=-500 and the target absolute position is PA=500, the PTP trajectory will travel through -500…-512, 511…500 to a total distance of 23 counts.

The cycle length must be positive (XM[2] > XM[1]) and (XM[2] - XM[1]) must be an even number. XM[N] values violating the above conditions will be accepted, but not activated. Until XM[N] can be activated, the motor cannot be started and the parameters cannot be saved in the flash memory (the SV command will not work).

Notes:

- If XM[1] or XM[2] is set so that XM[2] > XM[1] but PX is out of the range [XM[1]…XM[2]], PX will be set to range by taking modulo:
  PX = (PX - XM[1]) mod (XM[2] - XM[1]) + XM[1]
- If a homing process requests a PX value setting out of the range [XM[1]…XM[2]], the request will be ignored and PX will not change.
- If the XM[N] value is selected low and the main speed is too high, more than one full revolution of the main counter may elapse within a single sampling time. This will cause the main position counter to behave unpredictably.
- XM[N] cannot be modified while the homing sequence is active (HM[1] > 0).

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=0 MO=1 and SV cannot be used before XM[2] and XM[1] are activated (see "Activation" below)
	Default values:	XM[1]=-10^9, XM[2]=10^9 (RS), Non-volatile
	Range:	[-10^9…10^9]
	Unit modes:	All
	Activation:	XM[2] > XM[1] and XM[2] - XM[1] is an even number

See also:
YM[N], PX

XP[N] - Extra Parameters

Purpose:
Includes system parameters, not commonly changed by the user, which enable the drive to be adapted to special situations.

Index	Value Range	Meaning
0		Reserved.
1	[BV/16 … BV]	XP[1] defines over/under voltage threshold. Over voltage is activated at XP[1] volts. Under voltage is activated at XP[1]/8 volts.
2	[0…4]	PWM Fast Medium Slow mode. 0 – Medium switching frequency, PWM switch once per sampling time (TS) 1 – Slow switching frequency, PWM switches every two sampling time. 2..4- Fast switching frequency, PWM switches XP[2] times per sampling time
3		Reserved
4	[10…32,760]	Filter constant of bus voltage measurements.
5	[0…12,000]	Limiter for input filter of current loop.
6	[0…2,147,483,647]	Low-pass constant for input filter of current loop.
7	[0…8000]	Maximum counter value for Hall/encoder mismatch. If value is 0, no Hall/encoder mismatch check occurs.
8, 9		Reserved.

Table 2-41: **XP[N] Entries and Values**

Notes:

- Changing XP[1] or XP[2] requires a retuning of the current controller parameters.
- For the Cello and Trumpet under-voltage is activated at 0.18*BV
- When XP[2] is higher than 1, excessive PWM switching frequency occurs. In this case the drive's maximum allowed current saturation (PL[1]) is reduced. The amount of reduction depend upon the XP[2] value, the maximum bus voltage threshold setting (XP[1]), the type of drive, and the maximum allowed current (MC). To find out what the actual maximum stall current threshold is, use WS[33]. Use WS[34] to find out what the actual peak current saturation is.

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=0
	Default values:	XP[1]=BV, XP[2]=0, XP[4]=30720, XP[5]=1000 XP[6]=12000, XP[7]=0
	Range:	Previous table
	Unit modes:	All
	Activation:	Immediate

Reference chapter in the *SimplIQ Software Manual:*
Chapter 10, "The Current Controller"

See also:
BV

YA[N] - Auxiliary Position Sensor Parameters

Purpose:
Defines the behavior and direction of the auxiliary position sensor signals.

YA[4] can specify that the auxiliary encoder pins will be outputs, repeating the pulses of the main position incremental or analog sensors. When a main position feedback is by analog sensor, encoder emulation signals are produced. This mode is used to enable other drives to follow, without electrically overloading the main position sensor.

YA[4] can also specify that the encoder pins serve as input, following the pulses of the external position incremental sensor or PWM source signals. This mode is used to enable the drive to follow any source of encoder or PWM signals.

Index	Description
1	Emulation scale factor 2^N: N=[0...4] The amount of emulated encoder pulses and velocity value will be divided by 2^N.
2	Reserved.
3	Reserved.
4	Auxiliary encoder type: 0: Auxiliary encoder entry used as input for external pulse-and-direction signals 1: Not used 2: Auxiliary encoder entry used as input for external quadrature incremental encoder signals 3: Reserved 4: No auxiliary encoder inputs; the auxiliary encoder pin output repeats the main encoder input or produces an emulated encoder signal for drives with analog encoder or resolver feedback options. 5: 50% duty cycle command format. 0% and 100% duty cycle is interpreted as either maximum positive or negative PWM command depending upon the YA[5] setting. 6: 100% duty cycle command format. Depending upon command polarity signal level, 100% duty cycle is interpreted as either maximum negative or positive PWM command.
5	Auxiliary count direction: 0: Count forward for encoders or P&D inputs. For PWM inputs: forward direction, increasing the duty cycle causes an increase of the controller output command. 1: Count in reverse direction for encoders or P&D inputs. For PWM inputs: reverse direction, increasing duty cycle causes a decrease of the controller output command. YA[5] has no effect if YA[4] specifies that the auxiliary encoder pins repeat outputs (YA[4]=4).

Notes:

- Changing YA[4] resets the position sensor thus resetting the homing position.
- Changing YA[5] does not change PY. It only defines in which direction future encoder pulses will be counted.
- Changing the auxiliary sensor configuration while Output Compare is active may cause unpredictable drive behavior. No warning or error is given.
- For some Output Compare signal sources, the auxiliary sensor must be configured as output. Setting auxiliary sensor as input is illegal.

Attributes:	**Type:**	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=0
	Default values:	YA[1]…YA[3]=0 (RS), Non-volatile YA[4]=2, YA[5]=0 (RS), Non-volatile
	Range:	As in previous table
	Index range:	[1…5]
	Unit modes:	All
	Activation:	Immediate

See also: OC

YM[N] - Y Modulo

Purpose:
Specifies the counting range for the auxiliary feedback, which is [YM[1]…YM[2]-1].

The position of the auxiliary feedback is always counted cyclically. This means that after the position is counted to its maximum value, the next position count will reset the position counter back to its minimum value. The speed reading is not affected by the position jump. **For example:** If YM[1]=-5 and YM[2]=5, the auxiliary position is counted in a cycle length of 10. The auxiliary position will always be in the range [-5…4]. If the auxiliary feedback rotates in the positive direction, the auxiliary position count will proceed from 0, 1, 2, 3, 4 to -5, -4, -3, -2, -1, 0, 1 . . . and so on.

If YM[N] is nonzero with UM=4, the controlled motor position will count cyclically. All motion trajectories will go the short way. **For example:** If YM[1]=-512, YM[2]=512, the initial position is PY=-500 and the target absolute position is PA=500, the PTP trajectory will travel through -500…-512, 511…500 to a total distance of 23 counts.

The cycle length must be positive (YM[2]>YM[1]) and (YM[2] - YM[1]) must be an even number. YM[N] values violating the above conditions will be accepted, but not activated. Until YM[N] can be activated, the motor cannot be started and the parameters cannot be saved in the flash memory (the SV command will not work).

Notes:

- If YM[1] or YM[2] is set so that YM[2] > YM[1] but PY is out of the range [YM[1]…YM[2]], PY will be set to range by taking modulo:
  PY = (PY - YM[1]) mod (YM[2] - YM[1]) + YM[1]
- If a homing process requests a PY value setting out of the range [YM[1]…YM[2]], the request will be ignored and PY will not change.
- If the YM[N] value is selected low and the main speed is too high, more than one full revolution of the auxiliary counter may elapse within a single sampling time. This will cause the auxiliary position counter to behave unpredictably.
- YM[N] cannot be modified while the homing sequence is active (HM[1] > 0).

Attributes:		
	Type:	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	MO=0 MO=1 and SV cannot be used before YM[2] and YM[1] are activated (see "Activation" below)
	Default values:	YM[1]=-10^9, YM[2]=10^9 (RS), Non-volatile
	Range:	[-10^9…10^9]
	Unit modes:	All
	Activation:	YM[2] > YM[1] and YM[2] - YM[1] is an even number

See also:
XM[N], PY

ZP[N] - Integer Wizard Parameters

Purpose:
Contains parameters for internal use only.

Attributes:	**Type:**	Report, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Range:	Not applicable
	Index range:	[1…30]
	Unit modes:	All

ZX[N] - User Program and Auto-tuning Temporary Storage

Purpose:
Serves as temporary storage of reference waveforms for the controller in automatic tuning experiments. Out of the scope of auto-tuning experiments, the ZX[N] vector can be used as a large temporary storage for user programs.

Notes:

- The elements of ZX[N] are *short* (16-bit) integers.
- The ZX[N] command does not check input ranges. Assigning out-of-range values to ZX[N] can result in unpredictable results.

Attributes:		
	Type:	Parameter, Integer
	Source:	Program, RS-232, CANopen
	Restrictions:	None
	Default value:	0
	Range:	[-32,768…32,767]
	Index range:	[0…1023]
	Unit modes:	All
	Activation:	Immediate

Index

A

Abort · 2-89
Abort switch · 2-129
Absolute position · 2-33, 2-104
AC · **2-4**
Acceleration · 2-4
Activate recorder · 2-126
Activation · 2-2
Active current · 2-64
AG · **2-5**
AN · **2-6**
Analog
  Gains · 2-5
  Gains array · 2-5
  Input offsets array · 2-7
  Inputs array · 2-6
AOK · 2-96, 2-101, 2-102
Array
  Analog gains · 2-5
  Analog input offsets · 2-7
  Analog inputs · 2-6
  Commutation · 2-13
  GS[N] · 2-51
  Input bits · 2-63
  MP[N] · 2-93
  Output bits · 2-96
  QP · 2-93, 2-115
  QT · 2-93, 2-115
  QV · 2-93, 2-115
AS · **2-7**
Attributes · 2-1
Auxiliary
  Encoder · 2-42, 2-123, 2-161
  Feedback
    Position · 2-105, 2-118
    Velocity · 2-151
  Home
    Sequence · 2-68
    Switch · 2-67, 2-71, 2-73
  Homing and capture · 2-60
  Modulo · 2-163
  Position sensor · 2-98

B

Begin motion · 2-8, 2-11
BG · **2-8**
  Software speed mode · 2-8
  Stepper or position mode · 2-8
BH · **2-9**
Bit field · 2-1
BP · **2-10**
Brake · 2-96, 2-101, 2-102
Brake parameter · 2-10
BT · **2-11**
Burn
  User program · 2-27
BV · **2-12**

C

CA · **2-13**
CAN
  Communication · 2-110
  Controller status · 2-25
CANopen
  Communication · 1-2
  Parameters · 2-110
Capture
  Auxiliary · 2-60
  Main · 2-54
CC · **2-18**
CD · **2-19**
CL · **2-21**
Clear program · 2-23
Command sources · 1-1
Communication parameters · 2-110
Commutation
  Array · 2-13
  Setup · 2-15
  Tables · 2-58
Compiled program · 2-18
Composer · 2-27
Continue program · 2-157
CP · **2-23**
CPU
  Computing power · 2-142
  Dump · 2-19

Current
Continuous · 2-21
Control filter · 2-77
Limit flag · 2-80
Maximum peak · 2-107
Mode · 2-146

D

DC · **2-24**
DD · **2-25**
Deceleration · 2-24, 2-129
Default values · 2-2
Define recorded variables · 2-120
DF · **2-26**
Digital
Hall sensors · 2-13, 2-15
Input filter · 2-65
Input logic · 2-66
Output · 2-96
Flag · 2-54
Logic · 2-101
Uncommitted · 2-103
DL · **2-27**
Download
Firmware · 2-26
Program · 2-27
Dual loop · 2-146
DV · **2-28**

E

EC · **2-29**
ECAM
Parameters · 2-44
Table · 2-48
Echo mode · 2-46
EF · **2-42**
EM · **2-44**
Encoder · 2-14, 2-161
Auxiliary · 2-42, 2-123, 2-161
Filter frequency · 2-42
Glitch · 2-42
Main · 2-14, 2-42
Quadrature · 2-161
EO · **2-46**
ER · **2-47**
Error
Code · 2-29
Limit · 2-47
Errors
Motor failure · 2-86
Position · 2-105
Processing · 2-29
Velocity · 2-148
ET · **2-48**
Execute program · 2-157
Extra parameters · 2-156, 2-159

F

Feed forward · 2-49
FF · **2-49**
Filter
Current control · 2-77
Digital input · 2-65
Encoder · 2-42
High-order · 2-79
Velocity control · 2-77
Firmware
Download · 2-26
Version · 2-150
Flag
Auxiliary · 2-60
Main · 2-54
Floating Wizard parameters · 2-140
FLS · 2-67, 2-68, 2-71
Follower ratio · 2-50
Forward limit switch · 2-70
FR · **2-50**
Freewheel · 2-10, 2-69, 2-91
Frequency
Clock · 2-153
Encoder filter · 2-42
Sampling time · 2-121
Signal · 2-16

G

Gain
Analog signals · 2-5
Scheduled controller · 2-76
Scheduling · 2-51
General purpose · 2-68, 2-69, 2-101, 2-102, 2-103
GS · **2-51**

H

Halt program · 2-57
Hard
  Forward · 2-70
  Reverse · 2-69
  Stop · 2-69, 2-71
Hardware BG · 2-8
Hexadecimal · 2-9, 2-59
High
  Feedback limit · 2-53
  Reference limit · 2-149
High-order filter · 2-79
HL · **2-53**
HM · **2-54**
Home · 2-67
Home switch
  Auxiliary · 2-60, 2-67, 2-73
  Main · 2-54, 2-70, 2-73
Homing and capture
  Auxiliary · 2-60
  Main · 2-54
HP · **2-57**
HV · **2-58**
HX · **2-59**
HY · **2-60**

I

IB · **2-63**
ID · **2-64**
IF · **2-65**
IL · **2-66**
Inhibit · 2-69, 2-71, 2-102
Input
  Bits array · 2-63
  Filter · 2-65
  Logic · 2-66
  Port · 2-73
Integer · 2-1
Integer Wizard parameters · 2-164
Interrupt · 2-89
IP · **2-73**
IQ · **2-64**

J

Jogging
  Motion · 2-130
  Velocity · 2-75
JV · 2-8, 2-24, **2-75**

K

KG · **2-76**
KI · 2-76, **2-77**
Kill program · 2-78
KL · **2-78**
KP · 2-76, **2-77**
KV · **2-79**

L

LC · **2-80**
LD · **2-81**
Limit
  Flag (current) · 2-80
  High feedback · 2-53
  Low feedback · 2-82
  Switch
    Auxiliary Home · 2-71
    Forward · 2-70
    Main Home · 2-70
    Reverse · 2-69
Limits
  High and low reference · 2-149
  Position · 2-82
  Speed (velocity) · 2-82
  Torque (current) · 2-107
List
  Program · 2-84
  Properties · 2-83
LL · **2-82**
Load from flash · 2-81
Logic
  Input · 2-66
  Output · 2-101
Low
  Feedback limit · 2-82
  Reference limit · 2-149
LP · **2-83**
LS · **2-84**

M

Main
  Encoder · 2-14, 2-42
  Feedback · 2-158, 2-163

Position · 2-105, 2-117
Velocity · 2-151
Home
Sequence · 2-68
Switch · 2-54, 2-70, 2-73
Homing and capture · 2-54
Modulo · 2-158
Mask interrupt · 2-89
Maximum
Motor peak current · 2-107
Motor peak duration · 2-107
Peak driver current · 2-85
Position · 2-149
Position error · 2-47
Speed · 2-149
Tracking error · 2-47
velocity error · 2-47
MC · **2-85**
MF · 2-19, **2-86**
MI · **2-89**
Minimum
Position · 2-149
Speed · 2-149
Miscellaneous reports · 2-152, 2-153
MO · **2-91**
Modulo · 2-75, 2-141, 2-158, 2-163
Auxiliary · 2-163
Main · 2-158
Motion mode
JV · 2-8, 2-24, 2-75
PA · 2-8, 2-24
PR · 2-24
PT · 2-8, 2-93, 2-114, 2-119
PV · 2-8
PVT · 2-8, 2-93, 2-115, 2-119
Motion status · 2-95
Motor
Continuous phase current · 2-21
Disable · 2-91
Enable · 2-91
Failure · 2-86
Maximum peak current · 2-107
Read current · 2-64
Stuck protection · 2-21, 2-22
Voltage · 2-12
MP · **2-93**
MS · **2-95**

N

Non-volatile memory · 2-48

O

OB · **2-96**
OL · **2-101**
OP · **2-103**
Output
Bits array · 2-96
Logic · 2-101
Port · 2-103
Output configuration
AOK · 2-96, 2-101, 2-102
Brake · 2-96, 2-101, 2-102
General purpose · 2-101, 2-102, 2-103
Over-speed limit · 2-53

P

PA · 2-8, 2-24, **2-104**
Parameter · 2-1
PE · **2-105**
Peak duration and limit · 2-107
Peek Memory · 2-106
PI · 2-77
PK · **2-106**
PL · **2-107**
PM · **2-109**
Position
Absolute · 2-33, 2-104
Command · 2-28
Controller · 2-77, 2-79
Error · 2-105
Limits · 2-82
Mode · 2-146
Relative · 2-112
Position mode · 2-8, 2-75, 2-95
Position Time command · 2-114
Position Velocity Time command · 2-115
PP · 2-110
PR · 2-24, **2-112**
Profiled speed control mode · 2-24
Profiler · 2-109
Program
Compile · 2-18
Continue · 2-157
Download · 2-27

Execute · 2-157
Interrupt · 2-57, 2-78, 2-89
Status · 2-113
Upload · 2-84
Protection
Error limit · 2-47
Torque limit · 2-107
Protocol parameters · 2-110
PS · **2-113**
PT · 2-8, 2-93, **2-114**, 2-119
PTP · 2-104, 2-112, 2-130, 2-132
Pulse-and-direction mode · 2-161
PV · **2-115**
PVT · 2-8, 2-93, 2-119, 2-130
PX · **2-117**
PY · **2-118**

Q

QP · **2-119**
QP array · 2-93, 2-114, 2-115
QT · **2-119**
QT array · 2-93, 2-115
Quadrature · 2-42, 2-161
Quit
Abort · 2-89
Stop · 2-89
QV · **2-119**
QV array · 2-93, 2-115

R

Range · 2-2
RC · **2-120**
Reactive current · 2-64
Read current · 2-64
Real · 2-1
Record length · 2-122
Recorded variables · 2-128
Recorder
Activate · 2-126
Gap · 2-121
Get status · 2-126
Parameters · 2-124
Signals · 2-120
Triggers · 2-124
Reference
Commands · 2-28
Mode · 2-123
Relative position · 2-112
Reset
Counter · 2-42
Database · 2-33
Echo mode · 2-46
Hexadecimal mode · 2-59
Queue length · 2-35
Restrictions · 2-2
Reverse limit switch · 2-69
RG · **2-121**
RL · **2-122**
RLS · 2-67, 2-68, 2-69, 2-71
RM · **2-123**
RP · **2-124**
RR · **2-126**
RS · **2-127**
RS-232 · 1-1, 2-110
RV · **2-128**

S

Sampling time · 2-142
Save to flash · 2-136
Say something · 2-137
SD · **2-129**
Serial number · 2-131
SF · **2-130**
Smooth factor · 2-130
SN · **2-131**
Soft
Reset · 2-127
Stop · 2-71
Software
Speed mode · 2-8
Stop · 2-70
Source · 2-1
SP · **2-132**
Speed
Command · 2-28
Control mode · 2-146
Controller · 2-79
Limits · 2-82
Mode · 2-95
PTP · 2-132
Speed command · 2-69, 2-70
SR · 2-19, **2-133**
ST · **2-135**

Status
Register · 2-133
Report · 2-1
Stepper mode · 2-8, 2-75, 2-146
Stop · 2-63, 2-89
Deceleration · 2-129
Motion · 2-135
Stop manager · 2-28, 2-66, 2-130
String · 2-1
SV · **2-136**
Switch
Auxiliary Home · 2-71
Forward · 2-70
General purpose · 2-69
Main Home · 2-70
Reverse · 2-69
SY · **2-137**
System time · 2-139

T

Target radius · 2-141
TC · **2-138**
Temporary storage · 2-165
TM · **2-139**
Torque
Command · 2-28, 2-69, 2-70, 2-138
Control mode · 2-146
Limit · 2-107
TP · **2-140**
TR · **2-141**
Trigger · 2-124
TS · **2-142**
TW · **2-143**
Type · 2-1

U

UM · **2-146**
Unit mode · 2-2, 2-146
Dual loop · 2-2, 2-146
Position · 2-2, 2-146
Speed (velocity) · 2-2, 2-146
Stepper · 2-2, 2-146
Torque (current) · 2-2, 2-146
Upload program · 2-84
User program · 1-1

V

VE · **2-148**
Vector · **2-14**, 2-121, 2-152
Velocity
Command · 2-28
Control filter · 2-77
Error · 2-148
Feedback · 2-151
VH · **2-149**
VL · **2-149**
VR · **2-150**
VX · **2-151**
VY · **2-151**

W

WI · **2-152**
Wizard
Command · 2-143
Parameters
Floating · 2-140
Integer · 2-164
WS · **2-153**

X

X modulo · 2-158
XC · **2-157**
XM · **2-158**
XP · **2-156**, **2-159**
XQ · **2-157**

Y

Y modulo · 2-163
YA · **2-98**
YM · **2-163**

Z

ZP · **2-164**
ZX · **2-165**